

Technická univerzita v Liberci

---

**FAKULTA PŘÍRODOVĚDNĚ-HUMANITNÍ A PEDAGOGICKÁ**

---

**Katedra:** Katedra geografie

**Studijní program:** B1301 Geografie

**Studijní obor:** R022 Aplikovaná geografie

**Návrh zdokonaleného datového modelu pro síťové  
analýzy silniční sítě**

**Proposal of improved data model for road network  
analysis**

**Bakalářská práce:** 12-FP-KGE-026

**Autor:**

Jakub VIK

**Podpis:**

---

**Vedoucí práce:** Mgr. Jiří Šmída, Ph.D.

**Konzultant:**

**Počet**

stran	grafů	obrázků	tabulek	pramenů	příloh
68	0	29	11	13	6

V Liberci dne: 25.4.2012

TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta přírodovědně-humanitní a pedagogická  
Akademický rok: 2010/2011

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jakub Vik**  
Osobní číslo: **P08001056**  
Studijní program: **B1301 Geografie**  
Studijní obor: **Aplikovaná geografie**  
Název tématu: **Návrh zdokonaleného datového modelu pro síťové analýzy  
silniční sítě**  
Zadávající katedra: **Katedra geografie**

### Z á s a d y p r o v y p r a c o v á n í :

1. Rešeršní práce publikací věnujících se síťovým analýzám.
2. Vypracování metodiky a tvorba datového modelu s důrazem na přesný popis vertikálního a horizontálního průběhu komunikace.
3. Aplikace na modelovém území.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

**ARCTUR, D. - ZEILER, M. Designing Geodatabases: Case Studies in GIS Data Modeling. Redlands(California): ESRI, c1999-2004. ISBN 1-58948-021-X.**

**LONGLEY, P. A. - GOODCHILD, M. F. - MAGUIRE, D. J. - RHIND, D. W. Geographic Information Systems and Science. Chichester (England): Wiley and Sons, Ltd., c2001. ISBN 0-471-89275-0.**

**ORMSBY, T. - ALVI, J. Extending ArcView GIS. Redlands(California): ESRI, c1999. ISBN 1-879102-05-6.**


**TUČEK, J. Geografické informační systémy: principy a praxe. 1. vydání. Praha: Computer Press, c1998. ISBN 80-7226-091-X.**

Vedoucí bakalářské práce: **Mgr. Jiří Šmída, Ph.D.**  
Katedra geografie

Datum zadání bakalářské práce: **7. června 2011**  
Termín odevzdání bakalářské práce: **9. prosince 2011**

  
doc. RNDr. Miroslav Brzezina, CSc.  
děkan

L.S.

  
RNDr. Jaroslav Vávra, Ph.D.  
zástupce vedoucího katedry

V Liberci dne 30. března 2011

## Čestné prohlášení

**Název práce:** Návrh zdokonaleného datového modelu pro síťové analýzy silniční sítě

**Jméno a příjmení autora:** Jakub Vik

**Osobní číslo:** P08001056

Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména § 60 – školní dílo.

Prohlašuji, že má bakalářská práce je ve smyslu autorského zákona výhradně mým autorským dílem.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval/a samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Prohlašuji, že jsem do informačního systému STAG vložil/a elektronickou verzi mé bakalářské práce, která je identická s tištěnou verzí předkládanou k obhajobě a uvedl/a jsem všechny systémem požadované informace pravdivě.

V Liberci dne: 25.4.2012

---

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce Mgr. Jiřímu Šmídovi, Ph.D. za odborné a příkladné vedení po dobu tvorby práce. Poděkování patří také anonymním přispívatelům na geografických internetových fórech za jejich ochotu podělit se o zkušenosti s používáním GIS.

## ANOTACE

Bakalářská práce se zabývá implementací vhodných atributů do datového modelu silniční sítě, založeného na datové sadě ZABAGED, které by sloužily pro vyjádření horizontálního zakřivení komunikací. V úvodní teoretické části jsou popsány metody, kterými k dané problematice přistupují ostatní odborné práce. Na jejich základech je v praktické části navržen datový model, který za pomoci atributu úhlové změny a poloměru směrových oblouků lépe vyjadřuje skutečné vlastnosti komunikace. Postup tvorby modelu je přizpůsoben prostředí programu ESRI ArcGIS 10 s využitím skriptovacího jazyka Python. V závěru je navržený datový model použit pro analýzu dojezdových časů na modelovém území ORP Liberec. Na základě výsledků analýzy je ohodnocena funkčnost modelu a jsou navrženy další postupy pro jeho zpřesnění.

### KLÍČOVÁ SLOVA:

datový model, GIS, horizontální zakřivení, křivolakost, poloměr směrového oblouku, Python, silniční síť

## ANNOTATION

This thesis deals with the implementation of relevant attributes into the road network data model which is based on the Fundamental Base of Geographic Data; these attributes would be used to express the horizontal curvature of the road. Introductory theoretical part describes methods mentioned in other academic works. On this basis, the practical part of the thesis designs a data model which uses attributes of angular change and directional arc radius to reflect the real properties of roads. Process of designing the model is adapted to the environment of the ESRI ArcGIS 10 program using Python scripting language. In the end, the proposed data model is used for the purpose of the service area analysis in the territory of the administrative unit of Liberec. The functionality of the model is then evaluated on the basis of the results of the analysis and other potential ways of increasing the accuracy of the model are proposed.

### KEY WORDS:

data model, GIS, horizontal curvature, curvature index, directional arc radius, Python, road network

# OBSAH

Úvod .....	10
Cíle práce .....	11
1. Rešerše odborné literatury .....	12
2. Vysvětlení pojmů.....	16
2.1. Geografické informační systémy .....	16
2.2. Geografická data .....	16
2.3. Datové reprezentace, modely a formáty v GIS .....	17
2.4. ArcGIS .....	18
2.5. SQL.....	19
2.6. Python .....	19
3. Směrové řešení komunikace .....	20
3.1. Křivolakost.....	20
3.2. Poloměr oblouku.....	20
3.3. Druhy směrových oblouků.....	22
4. Skriptové nástroje .....	25
4.1. Skript 1 – Úhel a orientace .....	25
4.2. Skript 2 – Úprava koncových úhlů .....	33
4.3. Skript 3 – Řazení segmentů do oblouku .....	34
4.4. Newtonova metoda pro Field Calculator .....	36
5. Model pro hodnocení horizontálního zakřivení.....	37
5.1. Vstupní data .....	37
5.2. Teoretické předpoklady modelu .....	37
5.3. Struktura modulu .....	41

5.4. Vstup.....	42
6. Potvrzení funkčnosti modelu .....	47
6.1. Analýza průběhu komunikace.....	47
6.2. Rizikové úseky.....	48
6.3. Síťová analýza .....	50
7. Závěr.....	55
Literatura a použité zdroje .....	56
Seznam obrázků.....	57
Seznam tabulek .....	58
Seznam příloh .....	59



## **POUŽITÉ ZKRATKY A TERMÍNY**

AT – atributová tabulka

FID – (Feature Identification) unikátní kód prvku v atributové tabulce

GIS – geografický informační systém

IZS – integrovaný záchranný systém

ORP – obec s rozšířenou působností

ÚAP – územně analytické podklady

ÚÚP – úřad územního plánování

ZABAGED – Základní báze geografických dat (název datové sady)

## ÚVOD

Historie výstavby veřejné komunikační sítě je stejně stará jako historie stavby měst. Dopravování osob a nákladů byla vždy spjata s lidskou činností. Hovoříme-li konkrétně o silniční dopravě, je to dnes nejrychleji se rozvíjející odvětví dopravy na světě. Díky svým výhodám, zejména operativnosti a rychlosti, má rozhodující podíl na přepravě osob i nákladů. Díky vzrůstající kvalitě silniční sítě a dopravních prostředků začíná být využívána nejen pro vnitrostátní, ale v poslední době i pro mezistátní dopravu. Ve většině ekonomicky vyspělých států získala dokonce vedoucí postavení ve výkonu přepravy. S tím přichází i otázka bezpečnosti. Křivda (2007): *„Silniční doprava je nejméně bezpečným odvětvím dopravy a vyznačuje se nejvyšší nehodovostí. Denně zemře na silnicích ve světě 1100 lidí. Ročně při dopravních nehodách přijde o život 400 000 lidí, z toho 66 000 lidí zemře na evropských silnicích a 57 000 v USA, Kanadě a Japonsku. Ročně se při dopravních nehodách zraní kolem 12 mil. lidí.“* Silniční síť je také základním předpokladem pro pohyb jednotek vojska, policie a integrovaného záchranného systému (dále jen „IZS“). Z tohoto pohledu je zřejmé, že silniční síti je nutno věnovat velkou pozornost při jejím výstavbě, udržování a využití. K tomu se ve velké míře používají i tzv. GIS neboli geografické informační systémy. Jejich prostřednictvím lze síť nejrůznějším způsobem analyzovat a tím dosáhnout jejího lepšího využití. Například vhodně volenou analýzou komunikace lze určit místa se zvýšeným rizikem nehody. Dalším příkladem je analýza dojezdové doby vozidel IZS. Díky ní lze systém výjezdů jednotek optimalizovat a tím dosáhnout větší efektivity. Primárně je však zapotřebí připravit vhodná data, nad kterými se budou tyto i jiné analýzy provádět. Příprava dat pro účely síťových analýz je hlavním tématem práce.

## **CÍLE PRÁCE**

V rámci tématu bakalářské práce byly stanoveny tyto cíle:

1. Zhodnotit dosavadní metody hodnocení horizontálního zakřivení pozemních komunikací za použití geografických dat.
2. Navrhnout nový postup hodnocení, který by lépe odpovídal skutečným parametrům komunikace, a byl použitelný při síťových analýzách.
3. Na základě navrženého postupu vytvořit automatizovaný model pro platformu ArcGIS, který by pracoval s daty Základní báze geografických dat ČR.
4. Poukázat na problémy které se při vytváření modelu vyskytly a popsat metody, které vedly k jejich odstranění.
5. Na základě jednoduché síťové analýzy dokázat funkčnost navrženého modelu a výsledky prezentovat v grafickém výstupu.

# 1. REŠERŠE ODBORNÉ LITERATURY

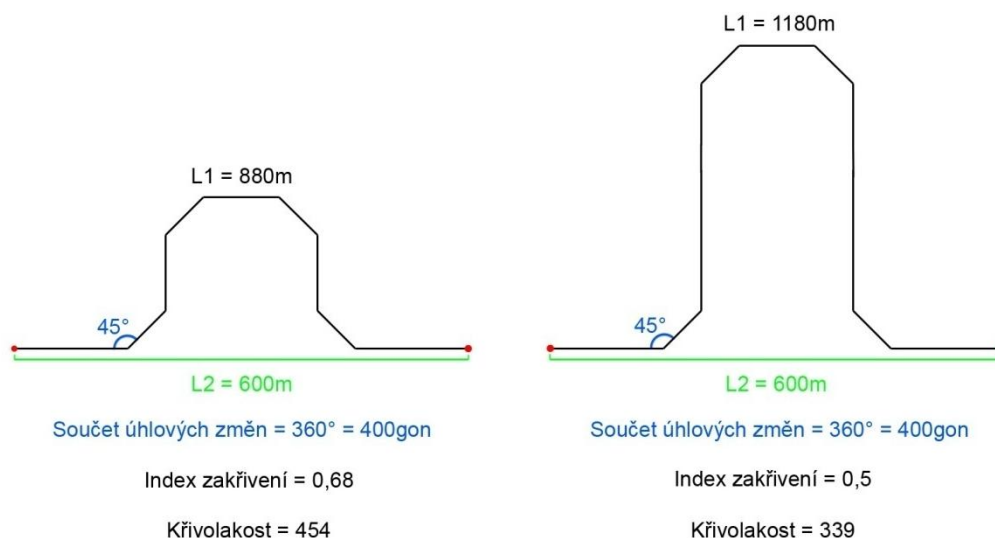
Jedním ze zdrojů týkajících se problematiky datových modelů pro síťové analýzy je ArcGIS Resource Center (Web-based 2011), což je webový portál pro uživatele softwaru ArcGIS od firmy ESRI. Tyto webové stránky obsahují veškeré informace o práci s programem, názorně vysvětlují jeho funkce a to formou textu i přehledných ilustrací. Tyto ilustrace byly v práci použity jako názorné příklady fungování některých použitých nástrojů. Zdroj také obsahuje velké množství návodů pro tvorbu datových modelů a práci s nimi. Ovšem pojem horizontálního zakřivení, jako faktoru ovlivňujícího průjezd komunikací, o kterém se zmiňuje Rasdorf (2011) a který je jedním ze základních pojmů pro bakalářskou práci, se v obsahu tohoto portálu nevyskytuje. Ani pro pojem index zakřivení (curvature index), se kterým pracuje Sladký (2009, s. 35), neexistují na portálu relevantní odkazy. Tento zdroj je proto v práci použit výhradně pro čerpání informací o funkcích programu ArcGIS.

ARCTUR D. a ZEILER M. v knize *Designing Geodatabases* a také BULTER J. A. v knize *Designing Geodatabases for Transportation* velmi shodně popisují stavbu a práci s geodatabázemi. Velmi podrobně popisují jednotlivé možnosti nakládání s geografickými daty a práci s rozsáhlými databázovými systémy. BULTER (2008) se dokonce zaměřuje výhradně na dopravní systémy. Tyto publikace jsou zdrojem pro teoretickou část práce zabývající se obecně prací s geografickými daty. Ovšem stejně také nepojednávají o horizontálním zakřivení komunikací.

Oficiální norma pro stavbu silnic a dálnic ČSN 736101 (2004) udává veškeré parametry pro stavbu komunikací. Pro horizontální zakřivení používá dva pojmy. V prvním případě se jedná o „křivolakost“, která charakterizuje geometrii osy komunikace. Jedná se o součet úhlových změn v gradech (gon) vztažený na délku posuzovaného úseku v kilometrech. Křivolakost je určována pro celé úseky komunikace. Pro hodnocení vlastností pro jednotlivé oblouky neboli zatačky, se v normě používá poloměr kružnicového oblouku (více v kapitole 3). Tato norma poskytla hlavní zdroj pro praktickou část bakalářské práce. Budou-li navrženým nástrojem vypočteny vzájemné úhly mezi přímkami tvořící linii komunikace. Bude možné jejich součtem určovat křivolakost úseků a tím také modifikovat atribut rychlosti průjezdu v následných analýzách.

Pojem zakřivení komunikace se objevuje v diplomové práci *Síťové analýzy v GIS pro složky IZS* (Sladký, 2009). Hlavním cílem práce je analýza dojezdových časů pro jednotky

integrovaného záchranného systému Plzeňského kraje a porovnání výsledků s naměřenými časy reálných výjezdů. V teoretické části je podrobně rozebrána problematika formátů geografických dat. Autor zde popisuje také formáty vektorových dat v jiných programech než je ArcGIS a to např. AutoCAD, Kokeš nebo Bantley Microstation. V praktické části popisuje autor faktory, které vstupují do analýzy dojezdové doby. Zmiňuje se zde o zakřivení komunikace. Zavádí pro něj pojem index zakřivení linie, který definuje takto: „Index zakřivení linie je číslo vyjadřující poměr mezi délkou linie vypočtenou ze souřadnic počátečního a koncového bodu linie a skutečnou délkou linie“ (Sladký, 2009, s. 46). V následné analýze používá tento index ke snížení rychlosti a to tak, že teoretickou rychlost vozidla násobí indexem zakřivení. Snižuje-li se tedy hodnota indexu, zvyšuje se náročnost průjezdu úsekem. Nezmiňuje se však o tom, jak jsou určeny úseky, pro které je index počítán. Z toho lze usuzovat, že délka úseků vychází z použitých dat, kterými jsou vrstvy komunikací ze ZABAGED. Úsek je, v případě těchto dat, část komunikace ležící mezi dvěma křižovatkami.



Obr. 1 Porovnání křivolakosti a indexu zakřivení

Obr. 1 jasně dokazuje závažný rozpor mezi křivolakostí (ČSN 736101 2004) a indexem zakřivení linie (Sladký, 2009). Jako příklad jsou dány dvě hypotetické linie komunikací se shodnými vzdálenostmi koncových bodů (L2) a shodným součtem úhlových změn. Liší se jen délkou linie (L1). Z příkladu je patrné, že prodloužením linie beze změny úhlů dojde ke snížení křivolakosti. Teoreticky se tedy zvýší průměrná rychlost při průjezdu. Naopak index zakřivení zaznamená pokles, což znamená snížení rychlosti. Index zakřivení

linie s takto určenými úseky má sice jistou vypovídající hodnotu o průběhu komunikace, je však výrazně nepřesný.

Další diplomovou prací, ve které figuruje zakřivení komunikace, jako faktor vstupující do síťové analýzy je Vztah digitálního modelu reliéfu a síťových analýz při řešení dopravních úloh (Louthan 2009). Autor v teoretické části práce cituje ČSN 736101 a uvádí, že směrodatná rychlost pro následnou analýzu je závislá na křivolakosti, kterou norma definuje. Dále v práci však uvádí: „*Dalším významným parametrem je zakřivení komunikace. ČSN ho definuje pojmem křivolakost a hraje rozhodující roli při stanovení finální směrodatné rychlosti. Řidič totiž musí vždy upravit rychlost jízdy v zatáčkách bez ohledu na ostatní faktory. V prostředí GIS by bylo velice složité vypočítat úhlové změny v každé zatáčce. Tento problém lze vyřešit výpočtem rozdílu mezi skutečnou délkou úseku a vzdušnou vzdáleností koncových bodů úseku*“ (Louthan 2009 s. 31).

Tab. 1 Určení směrodatné rychlosti (Louthan 2009, s. 37)

Návrhová rychlost (km/h)	Směrodatná rychlost v km/h při zakřivení		
	2 % a méně	2 – 5 %	5 % a více
15	25	25	25
20	30	30	30
30	40	40	35
35	45	45	40
40	50	50	45
50	60	60	55
55	65	65	60
60	70	70	60
65	75	70	60
70	80	80	70
75	85	80	75
80	90	80	80

Problém výpočtu zakřivení tedy řeší stejně jako (Sladký, 2009). Rychlost průjezdu úsekem stanovuje, na rozdíl od Sladkého, podle tabulky (Tab. 1), kde udává jako zdroj ČSN 736101 vydanou v roce 2000. V tomto zdroji se však taková tabulka, ani podklady pro ni nevyskytují. Pojem směrodatné rychlosti se v ČSN pro stavbu silnic a dálnic objevuje až v roce 2004 a je založen na křivolakosti a ne, jak autor v tabulce uvádí, na zakřivení (Tab. 1). Z těchto důvodů je možné se domnívat, že autor problém horizontálních změn směru komunikace obchází neurčitými postupy.

Tab. 2 Určení směrodatné rychlosti (ČSN 736101 2004)

Návrhová rychlost v km/h	Směrodatná rychlost v km/h při křivolakosti		
	≤ 249	250 - 315	316 a více
50	70	70	60
60	80	80	70
70	90	80	70
80	90	80	80
90	90	90	90

Z předchozích případů je patrné, že je nutné nalézt jiný způsob výpočtu zakřivení komunikací, než je porovnání délky linie a vzdálenosti koncových bodů. Zdrojem pro informace o dalších způsobech je článek z Journal of Computing in Civil Engineering, který publikoval v roce 2011 William Rasdorf, Ph.D., P.E profesor na North Carolina State University. Má název Evaluation of GIS Applications for Horizontal Curve Data Collection tedy v překladu Porovnání GIS aplikací pro sběr dat o horizontální křivce. Tento článek jako jediný, který byl nalezen, zmiňuje právě problém horizontálních křivek. Popisuje tři možné způsoby, jak měřit vlastnosti křivky v prostředí GIS.

Jednou je Curve Calculator, který je základní součástí programu ArcGIS a je obsažen v nástroji COGO (coordinate geometry). Jedná se o jakousi kalkulačku, která po manuálním zadání některých parametrů oblouku (zatačky) vypočítá její zbývající parametry. Výpočet poloměru provádí pouze pro kružnicové oblouky, neboť pro výpočet jiných typů křivek je třeba více parametrů. Například pro výpočet poloměru oblouku stačí zadat výšku a délku oblouku. Tyto hodnoty však lze určit teprve po identifikaci koncových bodů oblouku v průběhu linie.

Další popsanou možností je Curvature Extension, což je rozšíření pro ArcGIS 9 vyvinuté Florida Department of Transportation. Toto rozšíření obsahuje funkce, které dokážou automaticky získat informace o oblouku přímo z liniové vrstvy bez zadávání parametrů. Nevztahuje tedy veškeré výpočty na kružnicový oblouk a je tedy přesnější než Curve Calculator. Je však potřeba opět zadat začátek a konec oblouku. Tato extenze by byla velmi vhodná pro použití v práci, ovšem autorovi se nepodařilo ji pro tyto účely získat.

Třetí možností je Curve finder. Program, který vyvinul New Hampshire Department of Transportation. Jedná se o program, který, už podle názvu, dokáže automaticky vyhledat oblouky a následně určit jejich parametry. Jediné, co musí uživatel zadat, je vrstva,

koordinací systém a toleranci, která udává, jaký musí být odklon od přímky, aby byl oblouk identifikován. Tento program, jako jediný pracuje automaticky a je proto vhodný pro analýzy rozsáhlých sítí. Stejně však jako v případě předchozího nástroje se nepodařilo získat jej pro účely práce.

## **2. VYSVĚTLENÍ POJMŮ**

### **2.1. Geografické informační systémy**

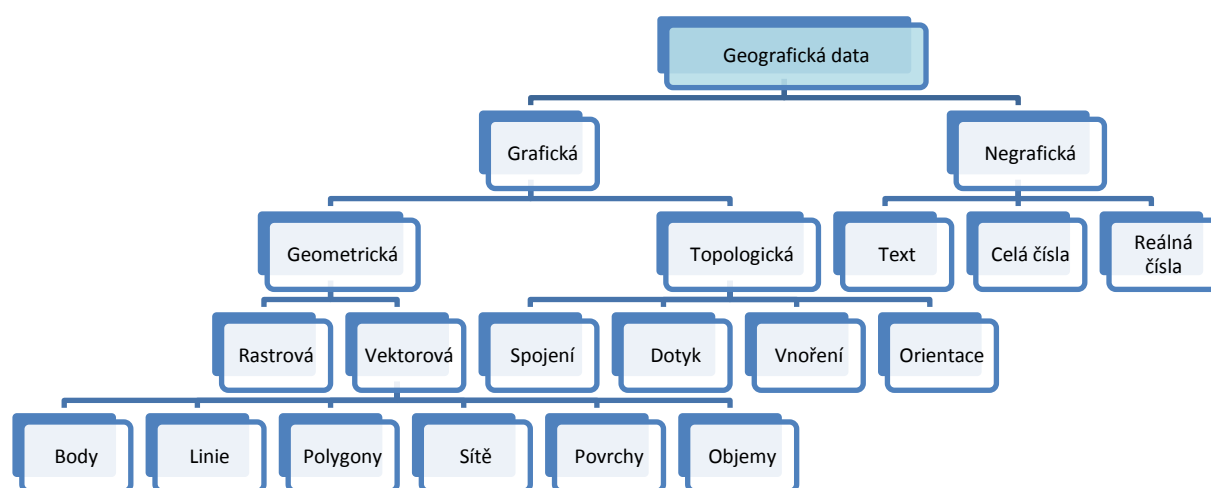
Jak bylo zmíněno v úvodu, práce je úzce spjata s pojmem geografický informační systém neboli GIS. Jedná se o pojem, který není v odborné literatuře definován vždy stejně. Například Tuček (1998) upozorňuje na to, že lze tento pojem chápat třemi možnými způsoby. Jednak se může jednat o samotný GIS software, tedy soubor programů pro správu a analýzu prostorových dat. V našem případě proto můžeme tvrdit, že se jedná o program ArcGIS 10 od společnosti ESRI, který je v práci využit. Pojem můžeme však chápat i jako soubor programů, zařízení, lidí a činností, který je součástí řízení určité instituce, jako je třeba okresní úřad. Ve třetí a nejobecnější rovině je pak GIS pojmenování pro celou vědní disciplínu. Ta přispívá k řešení obecných vědeckých problémů využíváním prostorových dat. Jednotnou a obsáhlejší definici uvádí společnost ESRI (2011): „GIS je organizovaný soubor počítačového hardwaru, softwaru a geografických údajů navržený na efektivní získávání, ukládání, upravování, obhospodařování, analyzování a zobrazování všech forem geografických informací“.

### **2.2. Geografická data**

At' přistoupíme k pojmu GIS jakkoli, je zřejmé, že základem jsou zde geografická data. Tato data jsou specifická tím, že obsahují nejen tematickou informaci, ale i záznam o své poloze v prostoru. Díky tomu je můžeme použít nejen k jejich lokalizaci, ale i k prostorovým analýzám. Voženílek (1998) uvádí rozdělení geografických dat (Obr. 2). Data se zde dělí na složku grafickou a negrafickou, kterou můžeme také nazvat tematickou. Ta popisuje pomocí textu, nebo číselných údajů kvalitu entity neboli základního prvku. Může také obsahovat klíč, kterým je prvek propojen v databázovém systému. Grafická složka obsahuje topologii, tedy informace o prostorových vztazích, které jsou buď uloženy přímo,



nebo se odvozují od geometrických dat. To záleží na způsobu uložení dat (více v kapitole 2.3). Geometrická část grafické informace je podstatou, která činí geografická data výjimečnými. Jsou kvantitativní povahy a slouží k vyjádření polohy entit v prostoru pomocí souřadnic. Geometrická data můžeme uchovávat buď v analogové formě jako tištěné mapy, nebo digitálně. Ukládáme-li data digitálně, musíme zvolit způsob jejich uložení, čili



Obr. 2 Dělení geografických dat (Voženílek 1998, upraveno)

reprezentaci. Tyto pak můžeme vkládat do složitějších schémat neboli datových modelů.

## 2.3. Datové reprezentace, modely a formáty v GIS

Přecházíme-li od reálného objektu k digitálnímu záznamu na paměťovém disku, musíme nejprve zvolit, jakým způsobem budou objekty v digitální formě reprezentovány. V prostředí GIS se nabízejí tři možnosti reprezentace: rastrová, vektorová a TIN (Triangulated Irregular Network).

### 2.3.1. Rastrová reprezentace

Podle Voženílka (1998) je elementární jednotkou rastrové reprezentace buňka. Podstatou rastru je překrytí zkoumané entity (objektu, nebo procesu) pravidelnou sítí, kde každé buňce v této síti přidělíme určitou hodnotu. Vznikne tak mřížka vytvářející pole (buňky), ve které má každá buňka svou danou polohu v prostoru na základě souřadnic a hodnotu, která zastupuje zobrazovaný jev. V praxi převládají rastry se stejnou rozlišovací úrovní, konkrétně s pravidelnou čtvercovou mřížkou. Ta se vyznačuje kompatibilitou

s maticemi, čili strukturami hojně využívanými ve výpočetní technice. Dá se na ně aplikovat systémy kartézských souřadnic a v neposlední řadě jsou podporovány zařízeními, jako jsou tiskárny a skenery. Kvalita reprezentace je závislá na rozlišení (velikosti buněk). Rastrová reprezentace je využívána především pro znázornění ploch a přechodů.

Rastrová reprezentace je v práci použita pro závěrečnou analýzu dojezdové doby. Analýza je provedena metodou „Cost Distance“ která pracuje na bázi rastrových dat.

### **2.3.2. Vektorová reprezentace**

Voženílek (1998) uvádí: „Vektorová data jsou posloupnosti bodů nebo linií, které jsou geometricky, nebo analyticky sloučeny. Toto sloučení znamená, že informace může být přiřazena bodu, nebo linii k vytvoření inteligentní množiny (vektoru), jenž obsahuje objemnější informaci, například obsahovou informaci ve formě více atributů.“ Vektorová data jsou vhodná pro reprezentaci takových objektů, které mají liniový charakter. Vodní toky, obvodové zdi budov, veškeré komunikace. Díky schopnosti věrně zachytit tvar a nést nejrůznější atributy jako údaj o směru průjezdu, třídě komunikace a sklonu jsou vektorová data vhodná pro síťové analýzy. V této práci se počítá s vektorovými daty, jako vhodnou reprezentací silniční sítě. Původní vektorová vrstva komunikace bude pomocí dále popsanych postupů obohacena o atributy, které umožní provádění nových síťových analýz.

## **2.4. ArcGIS**

Pro práci s geografickými daty se využívá software, který je speciálně uzpůsoben k tomu, aby pracoval s polohovou informací v datech obsaženou. V této době je na trhu hned několik programů např. GRASS GIS, Oracle Spatial, Quantum GIS a další. Pro účely práce byla použita platforma ArcGIS 10 vydávaná společností ESRI. Tento profesionální nástroj pro tvorbu a správu GIS je celosvětově rozšířen a podporován. V rámci Evropské unie tedy i České republiky se nástroj ArcGIS stává standardem v oblasti správy prostorových dat.

### **2.4.1. Model Builder**

Pro práci v prostředí ArcGIS 10 lze využít aplikaci „Model Builder“. Jedná se o nadstavbu pro vytváření úpravu a správu modelů. Modely jsou pracovní postupy, tedy

sekvence nástrojů, kde výstup jednoho nástroje stává se vstupem pro další. Slouží k automatizaci, tedy ke zjednodušení práce s daty.

## 2.5. SQL

„Structured Query Language“ v překladu strukturovaný dotazovací jazyk je soubor příkazů používaný k dotazování a manipulaci s daty v relačních databázových systémech (Esri 2011). ArcGIS používá SQL jako dotazovací jazyk pro spravované geodatabáze. Součástí mnoha nástrojů je požadavek na to, s jakými daty má nástroj pracovat. Tento požadavek se zadává právě v SQL.

## 2.6. Python

K vytvoření nových skriptů do programu ArcGIS 10 použitých jako nástroje v modelu pro výpočet horizontálního zakřivení, byl použit programovací jazyk Python. „Python je dynamický objektově orientovaný programovací jazyk, který v roce 1991 navrhl Guido van Rossum. Je vyvíjen jako open source projekt, který zdarma nabízí instalační balíky pro většinu běžných platform (Unix, Windows, Mac OS). Ve většině distribucí systému Linux je Python součástí základní instalace. Jazyk má vynikající vyjadřovací schopnosti. Kód programu je ve srovnání s jinými jazyky krátký a dobře čitelný“ (Harms in Louthan 2010, s. 42). „Výhodou aplikací napsaných v Pythonu je jejich výkon a v porovnání s jinými jazyky je na tom lépe. Pracuje až 5x rychleji než PHP. Python se snadno vkládá do jiných aplikací, kde pak slouží jako jejich skriptovací jazyk. Tím lze aplikacím psaným v kompilovaných programovacích jazycích dodávat chybějící pružnost. Jiné aplikace nebo aplikační knihovny mohou naopak implementovat rozhraní, které umožní jejich 43 použití v roli python modulu.“ (Beazley in Louthan 2010, s. 42, 43). Moduly jsou v podstatě textové soubory obsahující další příkazy ve stejném, nebo i jiném programovacím jazyce. Na tomto principu funguje Python v prostředí ArcGIS. Slouží v něm jako skriptovací jazyk, který umožňuje uživateli rozšířit množství funkcí, kombinovat a automatizovat různé nástroje. Podle Lutze (1999, s. 4-7) jsou hlavním důvodem oblíbenosti Pythonu tyto jeho vlastnosti. Je objektově orientovaný, což znamená, že objekty, jako entity programu se navenek chovají jako samostatné jednotky, pamatují si svůj stav a navzájem spolu komunikují. Je zdarma dostupný a lehce přenositelný na jakoukoli platformu. Je výkonný a umí spolupracovat s celou škálou ostatních jazyků. Posledním důvodem je, že se snadno učí.

Při práci na nových skriptech byl kód sestavován v programu PythonWin, který vyvinul Mark Hammond. Jedná se o textový editor pro Windows přizpůsobený přímo k práci s jazykem Python. Obsahuje mnoho funkcí, které zrychlují a zpřehledňují sestavování skriptu.

### 3. SMĚROVÉ ŘEŠENÍ KOMUNIKACE

V rešeršní části byly popsány některé způsoby, jak lze z dat získat informace o zakřivení komunikace. Bylo však prokázáno, že jsou značně nepřesné a zavádějící. Pro navržení postupu při hodnocení zakřivení komunikace je třeba vycházet z reálného tvaru komunikace, konkrétně z vedení její osy.

#### 3.1. Křivolakost

Hodnota zakřivení celých úseků komunikace se vyjadřuje křivolakostí. Jedná se o hodnotu, která se podílí na určení směrodatné rychlosti. Jedná se o součet úhlových změn závislý na délce úseku. Křivolakost je vyjádřena vzorcem:

$$K = \frac{\sum_{i=1}^j \gamma_i}{l}$$

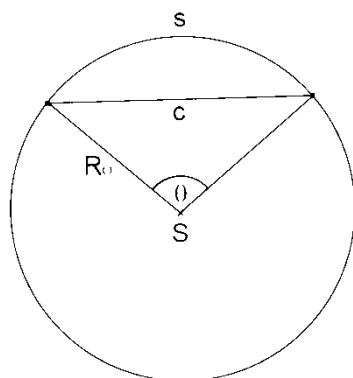
kde  $K$  je křivolakost,  $\gamma_i$  je úhlová změna vyjádřená v gradech (1gon = 0,9stupňů) pro jednotlivé úseky a  $l$  je pak délka všech úseků (ČSN 736101, 2004, s. 22).

Navrhovaný nástroj je schopen určit velikosti úhlů mezi jednotlivými segmenty linie. Díky tomu lze v libovolném úseku komunikace určit jeho křivolakost. Na základě křivolakosti úseků komunikace je provedena jedna ze zkušebních analýz. Výsledky analýzy jsou uvedeny v příloze 5.

#### 3.2. Poloměr oblouku

Pro určení vlastností jednotlivých oblouků (zataček) v průběhu komunikace se nejlépe hodí hodnota udávající poloměr oblouku v metrech. Tab. 9, která je součástí normy pro stavbu silnic a dálnic, jasně dokazuje, že poloměr kružnicového oblouku je určující hodnotou pro rychlost průjezdu obloukem. Poloměr oblouku je použit pro úpravu výsledné rychlosti pro zkušební analýzu v závěru práce. Výsledky analýzy jsou uvedeny v příloze 6.

### 3.2.1. Určení poloměru oblouku

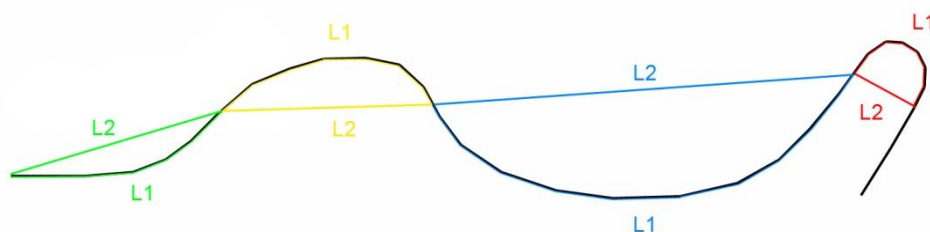


$$c/s = \sin(x)/x$$

$$\theta = 2x$$

$$R_o = s/\theta$$

Obr. 4 Kruhová úseč



Obr. 3 Určení oblouků v průběhu linie

Při určování poloměru kružnicového oblouku z dostupných dat (ZABAGED) je v případě modelu použit postup, který vychází z hodnot skutečné délky linie (L1) a vzdálenosti koncových jejích koncových bodů (L2). Tyto hodnoty určuje pro úseky, které odpovídají jednotlivým obloukům, jak je znázorněno na Obr. 3. Díky správnému určení koncových bodů oblouku je možné přejít na výpočet jeho poloměru. K tomuto účelu je využita tzv. Newtonova metoda. Jedná se numerickou metodu pro řešení nelineárních rovnic. Nazývá se také metoda tečen. Pomocí této metody lze vypočítat poloměr kružnice, známe-li tětivu výseče  $c$  a délku oblouku  $s$ . Na Obr. 4 je znázorněna kruhová úseč pro upřesnění, jaké hodnoty vstupují do výpočtu poloměru. Výpočet poloměru probíhá tak, že hledáme  $x$  pro které platí  $c/s = \sin(x)/x$ . Poloměr oblouku ( $R_o$ ) se vypočítá jako  $R_o = s/\theta$  kde  $\theta = 2x$ . Úhel  $\theta$  je určen radiánech.

## Newtonova metoda

Tato kapitola je zpracována podle příspěvku na matematickém fóru (Segments of Circles 2012). Pro určení poloměru je nutné nalézt  $x$  pro které platí  $\frac{\sin(x)}{x} = k$ , kde  $k$  je kladná konstanta určená tětivou a obloukem  $c/s = k$ . Hledáme tedy kořen rovnice  $\sin(x) - kx = 0$ . Pro nalezení  $x$  slouží právě Newtonova metoda. Je dáno  $x_0 = \sqrt{6 - 6k}$ , dále pro každé  $n = 1, 2, 3, \dots$  (řešíme rovnici  $x_{n+1} = x_n - (\sin x_n - kx_n)/(\cos x_n - k)$ ). Počet  $n$  je dán počtem desetinných míst výsledku. Jako příklad stanovme hodnotu  $k = \frac{3}{4}$  a hledejme  $x$  s přesností na pět desetinných míst. Jakmile  $|x_{n+1} - x_n| < 0,000005$  získáme výsledek s požadovanou přesností  $x = 1,27570$ . Úhel odpovídá dvojnásobku  $x$ , tedy  $\theta = 2,5514 \text{ rad}$ . V případě, že oblouk má délku 400m je výsledný poloměr 156,77667m.

Tab. 3 Část výpočtu Newtonovou metodou (Segments of Circles 2012)

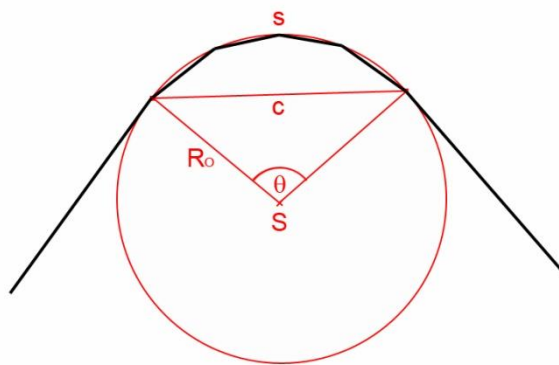
n	x (n)	sin (x [n])	cos (x [n])	sin (x [n]) / x (n)
0	1.2247449	0.9407193	0.3391860	0.7680941
1	1.2786882	0.9576389	0.2879717	0.7489229
2	1.2757074	0.9567763	0.2908250	0.7499967
3	1.2756981	0.9567736	0.2908338	0.7500000
4	1.2756981			

### 3.3. Druhy směrových oblouků

Kružnicový oblouk je sice základním tvarem užívaným v geometrii komunikací, není však jediným ČSN 736101 (2004) definuje 4 druhy směrových oblouků. Druh oblouku má vliv na výslednou přesnost při výpočtu poloměru oblouku.

### 3.3.1. Prostý kružnicový

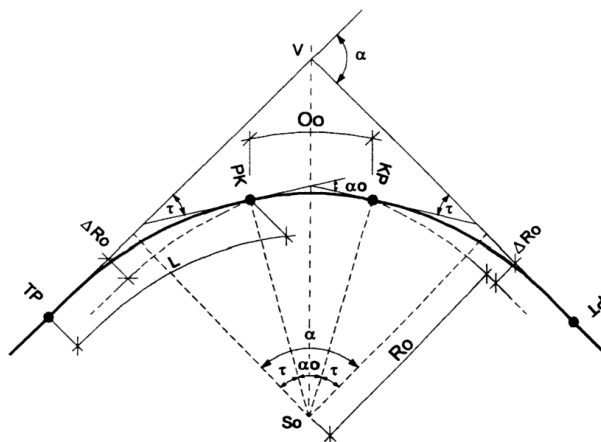
Jedná se o kružnicový oblouk vložený mezi dva rovné úseky. Při výpočtu poloměru nevzniká žádná chyba, neboť nástroj počítá právě s oblouky kružnicovými. Na Obr. 5 je zobrazen příklad výpočtu aplikovaný na linii komunikace (černá linie).



Obr. 5 Oblouk prostý kružnicový

### 3.3.2. Kružnicový s přechodnicemi

Jedná se o nejčastější řešení směrového oblouku. Skládá se z kružnicové části a zpravidla z oboustranných přechodnic. Na Obr. 6 se kružnicový oblouk nachází mezi body PK a KP. Přechodnice jsou křivky mezi body TP a PK, KP a PT. Přechodnice je křivka, která plynule mění svou křivost. Používá se ke snížení příčného rázu, který vzniká při přechodu z přímky s nulovou křivostí na kružnicový oblouk, kde je křivost konstantní hodnota odvozená z poloměru. V silničním stavitelství se běžně používá jako přechodnice křivka zvaná klotoida. Klotoida je nekonečně dlouhá křivka, která lineárně mění svoji křivost.



Obr. 6 Oblouk kružnicový s přechodnicemi (ČSN 736101 2004)

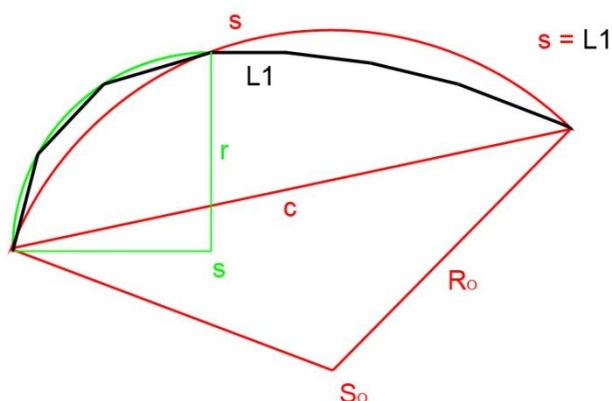
Křivost  $p$  je inverzní hodnota poloměru  $R$  dána rovnicí  $p = \frac{1}{R}$ . Základní rovnice klotoidy je pak  $l \times R = A^2$ , kde  $l$  je délka klotoidy a  $A$  je konstanta udávající její velikost.

V případě tohoto druhu oblouku je při použití nástroje pro výpočet poloměru nutné počítat s určitou chybou. Důvodem je skutečnost, že oblouk s přechodnicí neodpovídá kružnici. Vypočteme-li tedy kružnicový oblouk s přechodnicemi jako prostý kružnicový oblouk, bude výsledný poloměr větší, než je poloměr kružnicové části oblouku. Rozdíl poloměrů závisí na tom, jaký díl z celkové délky oblouku zaujímají přechodnice. Norma udává doporučené délky přechodnic  $L$  v závislosti na poloměru kružnicového oblouku  $R_o$ . Doporučené délky přechodnic jsou uvedeny v Tab. 4.

Tab. 4 Doporučené délky přechodnic (ČSN 736101 2004)

$R_o$ v m	100	200	300	500	1 000	1 500	2 000	3 000	4 000	5 000
$L$ v m	60	80	100	120	160	210	290	430	500	550

Na Obr. 7 je znázorněn posun středu a změna poloměru při převodu kružnicového oblouku s jednou přechodnicí na oblouk prostý kružnicový při zachování délky oblouku  $s$ . Linie osy komunikace  $L1$  (černá) odpovídá svojí délkou oblouku  $s$ . Červeně jsou naznačeny hodnoty, se kterými počítá nástroj na výpočet poloměru. Zeleně je pak naznačena kružnicová část oblouku. Nástroj tedy při výpočtu zahrnuje přechodnici, jako by byla součástí kružnicového oblouku.



Obr. 7 Převod oblouku kružnicového s přechodnicí na prostý kružnicový



### 3.3.3. Složený

Složeným obloukem se míní oblouk, který se skládá z více kružnicových oblouků s různými poloměry. Používá se především při návrhu křižovatek. Při použití nástroje vzniká podobná chyba jako u předchozího případu. Chyba je v tomto případě závislá na rozdílu poloměrů dílčích oblouků, ze kterých se celkový oblouk skládá.

### 3.3.4. Přechodnicový

Tento druh oblouku tvoří dvě navazující přechodnice. Oblouk tedy neobsahuje část s konstantním poloměrem. Lze však z rovnice klotoidy určit maximální poloměr, kterého přechodnice v daném oblouku dosahuje. Chyba při výpočtu s použitím navrhovaného modelu je závislá na konstantě A, která udává parametry klotoidy. Moderní přístupy ke stavbě silnic a dálnic se čím dál více přiklánějí k tomuto druhu oblouků. Důvodem je skutečnost že přechodnicový oblouk má plynulou změnu křivosti a nejlépe splňuje také požadavky estetické.

## 4. SKRIPTOVÉ NÁSTROJE

Pro splnění cílů práce byly autorem navrženy nové nástroje do prostředí ArcGIS 10, které jsou následně zakomponovány do struktury výsledného modelu. Nástroje jsou založeny na skriptech v jazyce Python. Texty všech skriptů jsou obsaženy v přílohách 1, 2 a 3. V této kapitole jsou skripty podrobně rozebrány a funkce jednotlivých příkazů je zde vysvětlena jednak v odstavci pod textem skriptu, nebo také v poznámkách přímo v textu skriptu umístěných za znakem „#“. Tento znak je užíván v jazyce Python pro psaní poznámek, které na průběh skriptu nemají vliv.

### 4.1. Skript 1 – Úhel a orientace

Tento skript je použit v modelu pro výpočet horizontálního zakřivení. Na základě souřadnic začátečních a koncových bodů jednotlivých liniových prvků (segmentů) vstupní vrstvy, vypočítá úhlovou změnu mezi jednotlivými segmenty a její orientaci. Hodnoty následně zaznamená do nově vytvořených polí v atributové tabulce dané vrstvy.

## Podrobný popis skriptu:

```
import arcpy, math
originalinputfeature = arcpy.GetParameterAsText(0)
outputfeature = arcpy.GetParameterAsText(1)
arcpy.CopyFeatures_management(originalinputfeature, outputfeature)
arcpy.AddField_management(outputfeature,"UhelMinus","DOUBLE","","","","UhelSegment-1")
arcpy.AddField_management(outputfeature,"UhelPlus","DOUBLE","","","","UhelSegment+1")
arcpy.AddField_management(outputfeature,"DirectionMinus","TEXT","","","")
arcpy.AddField_management(outputfeature,"DirectionPlus","TEXT","","","")
```

Importuje jednotlivé moduly, které jsou vyžadovány pro provádění dalších příkazů. Definuje proměnné pro vstupní a výstupní vrstvu. Jejich hodnoty načítá z dialogového okna při zpuštění nástroje. Funkcí „CopyFeatures“ kopírováním vytvoří výstupní vrstvu, kterou obohatí o čtyři nová pole v atributové tabulce. Jejich vlastnosti popisuje Tab. 5.

Tab. 5 Vlastnosti nových polí AT

NÁZEV POLE	DATOVÝ TYP	ALIAS	ROZSAH HODNOT (v rámci nástroje)	POPIS
UhelMinus	DOUBLE	UhelSegment-1	0 až 90	Úhel se segmentem, jehož FID je o jednu menší.
UhelPlus	DOUBLE	UhelSegment+1	0 až 90	Úhel se segmentem, jehož FID je o jednu větší.
DirectionMinus	TEXT		Right, left, 0	Směr stáčení oblouku
DirectionPlus	TEXT		Right, left, 0	Směr stáčení oblouku

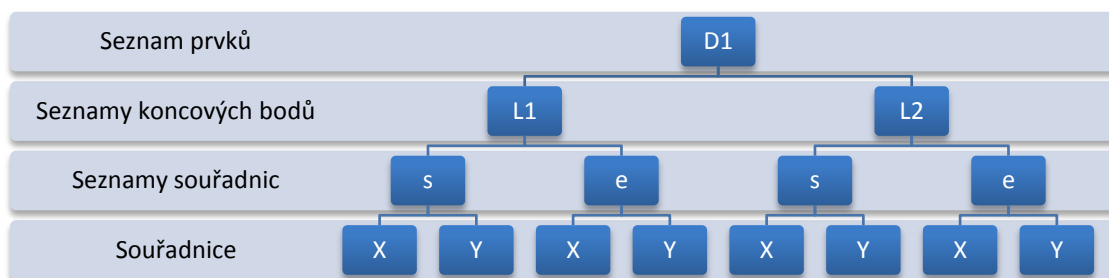
```
desc = arcpy.Describe(outputfeature)
cur = arcpy.SearchCursor(outputfeature)
```

Načte informace o vrstvě do objektu „desc“ (describe). Vytvoří kurzor pro načítání hodnot z atributové tabulky. Jedná se o souřadnice koncových bodů jednotlivých segmentů obsažené v poli „Shape“.

```
s=[0.0,0.0]          #vytvoř seznam s párem nulových souřadnic
e=[0.0,0.0]          #vytvoř seznam s párem nulových souřadnic
l=[s,e]              #vytvoř nadřazený seznam
D = [l]              #vytvoř seznam s prvkem „l“
```

Pro uložení načtených souřadnic je vytvořen víceúrovňový seznam, kde „s“ (start) a „e“ (end) reprezentují dvojice souřadnic koncových bodů segmentu (přímky). Nadřazeným seznamem je pak „D“ (data), který obsahuje seznam jednotlivých „l“ (line). Obsahem každého prvku v „l“ je právě dvojice koncových bodů. Struktura seznamu je znázorněna na Obr. 8. Pro následné načítání hodnot z takto vytvořeného víceúrovňového seznamu lze použít

jednoduchý postup, při kterém je zadáno vždy pořadí prvku v úrovni seznamu. Např. pro načtení hodnoty pro y souřadnici začátečního bodu druhého prvku souží příkaz  $Y=D[2][1][2]$  (druhý prvek v seznamu prvků, první prvek v seznamu koncových bodů a druhý prvek v seznamu souřadnic).



Obr. 8 Struktura víceúrovňového seznamu

```
for row in cur:
    feat=row.shape                #načti hodnoty z pole „Shape“
    firstpoint = feat.firstPoint
    lastpoint = feat.lastPoint
    startx = firstpoint.X
    starty = firstpoint.Y
    endx = lastpoint.X
    endy = lastpoint.Y
    s=[startx,starty]
    e=[endx,endy]
    l=[s,e]
    D.append (l)                 #do seznamu „D“ přidej položku obsahující „l“
del row
del cur
```

Tato část skriptu je příkazem pro definovaný kurzor „cur“, který projde všechny řádky AT a provede zapsání souřadnic do seznamu. Výsledkem je kompletně vyplněný seznam pro všechny segmenty.

```
beta = 0
direction = "0"
startx1 = D[1][0][0]
starty1 = D[1][0][1]
endx1 = D[1][1][0]
endy1 = D[1][1][1]
```

Před začátkem výpočtu pro všechny segmenty linie je třeba určit výchozí hodnoty pro první segment. Úhel je zatím nulový, protože první segment nemá předchůdce, podle kterého by úhel počítal. S nulovým úhlem je spojena i nulová orientace.

```
cur = arcpy.UpdateCursor (outputfeature)          #kurzor pro editaci atributů
for row in cur:
    FID = row.getValue(desc.OIDFieldName)          #FID aktivního řádku
    row.UhelMinus = beta                           #vepiš „beta“ do pole „UhelMinus“
    row.DirectionMinus = direction
    try:
        del beta, direction, trend1, trend2, alfa1, alfa2
    except:
        print "beta atd. is not deleted"
    beta = []
```

Po nastavení kurzoru začne opakování pro jednotlivé řádky. Načtením informace o FID získáme údaj, v jakém řádku AT se kurzor nachází. Do polí označených „Minus“ jsou vepsány hodnoty „beta“ a „direction“. Poté mohou být tyto hodnoty smazány, aby mohly být nahrazeny novými.

```
try:
    startx2=D[FID +1][0][0]
    starty2=D[FID +1][0][1]
    endx2=D[FID +1][1][0]
    endy2=D[FID +1][1][1]
except:
    startx2=D[FID][0][0]
    starty2=D[FID][0][1]
    endx2=D[FID][1][0]
    endy2=D[FID][1][1]
```

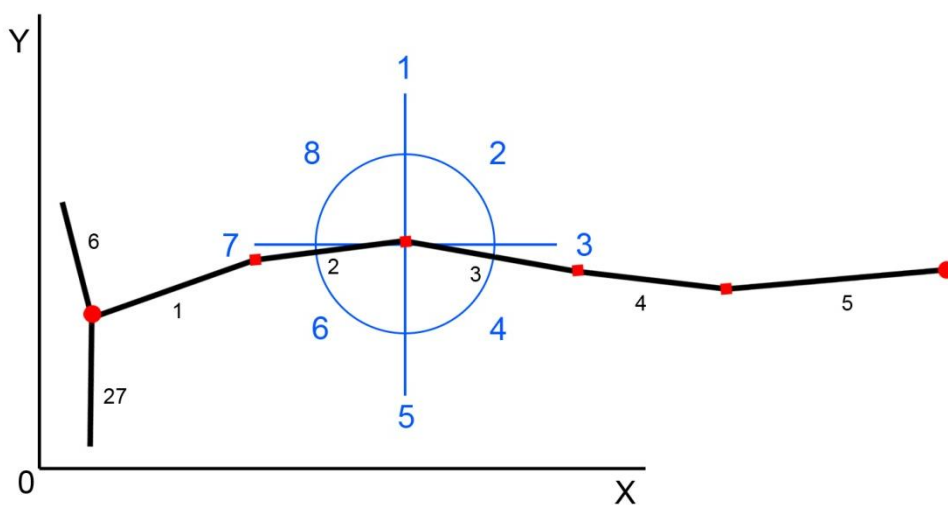
Nyní se načítají hodnoty pro následující segment. Ze seznamu „D“ je vybrán prvek s číslem o jednu vyšším, než je aktuální hodnota FID. Operace je pod příkazem „try“ z toho důvodu, aby u posledního řádku AT, kde se již následující údaje v seznamu nevyskytují, nehlásil skript chybu. V případě posledního segmentu načte skript totožné hodnoty a tak výsledek pro poslední segment bude nulový.

```

if startx1==endx1 and starty1<endy1:
    trend1=1
elif startx1!=endx1 and starty1!=endy1 and startx1<endx1 and starty1<endy1:
    trend1=2
elif starty1==endy1 and startx1<endx1:
    trend1=3
elif startx1!=endx1 and starty1!=endy1 and startx1<endx1 and starty1>endy1:
    trend1=4
elif startx1==endx1 and starty1>endy1:
    trend1=5
elif startx1!=endx1 and starty1!=endy1 and startx1>endx1 and starty1>endy1:
    trend1=6
elif starty1==endy1 and startx1>endx1:
    trend1=7
else:
    trend1=8

```

V této části je stanoven tzv. „trend“. Nabývá hodnot od 1 do 8 a jedná se o vlastnost udávající orientaci přímky v pravoúhlé soustavě souřadnic při pohledu z jejího začátečního bodu. „Trend“ vychází z rozdílu souřadnic začátečního a koncového bodu segmentu. Ve skriptu je použito několika po sobě jdoucích podmiňovacích příkazů. Nemá-li splněna první podmínka, ve které se vzájemně porovnávají souřadnice bodů, přestoupí se na druhou podmínku atd. Je-li však podmínka splněna, ostatní příkazy se neprovádějí. Na Obr. 9 je graficky znázorněno, jak jsou hodnoty pro „trend“ stanoveny. V případě, že je segment rovnoběžný s některou z os, nabývá „trend“ lichých hodnot. Sudé hodnoty pak zastupují jednotlivé kvadranty. V tomto příkladu je pro segment 3 „trend“ = 4.

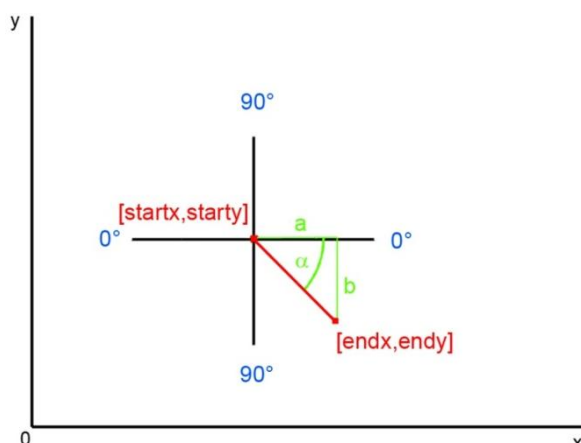


Obr. 9 Stanovení trendu

```

if trend1 in [1,5]:
    alfa1=90.0
elif trend1 in [3,7]:
    alfa1=0.0
elif trend1 in [2,4,6,8]:
    x1=abs(startx1-endx1)
    y1=abs(starty1-endy1)
    alfa1=math.degrees(math.atan(y1/x1))

```



$$a = |startx - endx|$$

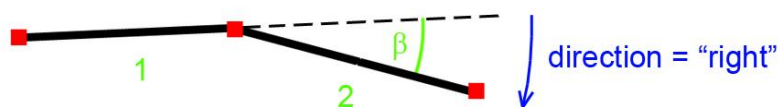
$$b = |starty - endy|$$

$$\alpha = \tan^{-1}(b/a)$$

Obr. 10 Výpočet úhlu k ose x

Po zjištění trendu lze přistoupit k výpočtu velikosti úhlu k ose x. Tento úhel nazvaný ve skriptu „alfa“ společně s vlastností „trend“ je výchozí hodnotou pro výpočet úhlové změny mezi jednotlivými segmenty linie. Pokud „trend“ nabývá lichých hodnot je segment k ose x kolmý, či s osou rovnoběžný. V případě sudých hodnot se nejprve ze souřadnic koncových bodů určí vzdálenosti na osách. Poté se úhel vypočítá pomocí funkce tangens, jak je naznačeno na Obr. 10.

Následující část skriptu není v textu uvedena, neboť obsahuje totožné příkazy pro určení trendu a výpočet úhlu k ose x. Pracuje však se souřadnicemi druhého, tedy následujícího segmentu. Výsledkem je tedy úhel k ose x pro dva po sobě jdoucí segmenty linie (alfa1, alfa2) a jejich orientace definovaná hodnotou „trend“ (trend1, trend2). Následuje výpočet úhlové změny (beta). Na Obr. 11 je znázorněna úhlová změna dvou segmentů. Pokud jsou segmenty rovnoběžné, úhel beta je nulový.



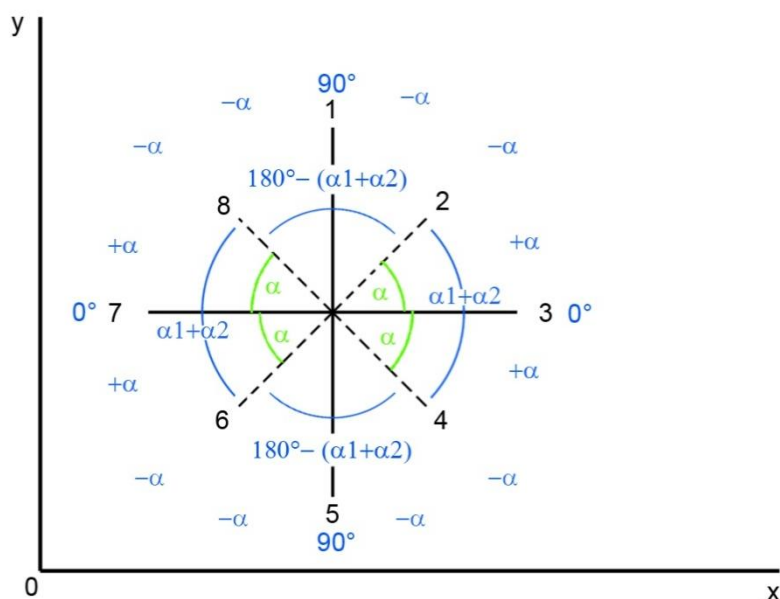
Obr. 11 Úhlová změna s orientací

```

if trend1==trend2:
    beta=abs(alfa1-alfa2)
elif trend1 in [1,3,5,7] and trend2 in [1,3,5,7]:
    beta=abs(alfa1-alfa2)
elif abs(trend1-trend2) in [0,1,2,6,7]:
    if (trend1 in [2,4] and trend2 in [6,8])or(trend1 in [6,8] and trend2 in [2,4]):
        beta=abs(180.0-(alfa1+alfa2))
    elif (trend1 in [8,2] and trend2 in [4,6])or(trend1 in [4,6] and trend2 in [8,2]):
        beta=abs(alfa1+alfa2)
    elif (trend1 in [2,4] and trend2 in [2,4])or(trend1 in [6,8] and trend2 in [6,8]):
        beta=abs(alfa1+alfa2)
    elif (trend1 in [1,5] and trend2 in [2,4,6,8])or(trend1 in [2,4,6,8] and trend2 in [1,5]):
        beta=abs(alfa1-alfa2)
    elif (trend1 in [3,7] and trend2 in [2,4,6,8])or(trend1 in [2,4,6,8] and trend2 in [3,7]):
        beta=abs(alfa1+alfa2)
    else:
        beta = 90.0
else:
    beta = 90.0

```

Pro výpočet úhlové změny (beta) je třeba určit, v jakém vztahu jsou hodnoty „trend“ obou segmentů. Na základě tohoto vztahu je pak zvolen postup výpočtu. Pro některé vztahy je nutné úhly „alfa“ odečítat, pro některé sčítat a některé vyžadují odečíst jejich součet od 180°.



Obr. 12 Postup výpočtu úhlu beta pro jednotlivé kombinace trendů

Veškeré vztahy a postupy výpočtu jsou znázorněny na Obr. 12. Ve skriptu se výpočet skládá z jednotlivých logických podmínek, které postihují veškeré kombinace hodnot „trend“, které nastanou, je-li úhlová změna menší nebo rovna  $90^\circ$ . S úhly nad  $90^\circ$  nástroj nepočítá, neboť takové úhly se reálně vyskytují pouze v rámci křížení. Nástroj veškeré úhly větší než  $90^\circ$  vyhodnocuje jako úhly pravé.

```

if trend1==trend2 and alfa1==alfa2:
    direction="0"
elif trend1==trend2 and trend1 in [2,6]:
    if alfa1>alfa2:
        direction="right"
    if alfa1<alfa2:
        direction="left"
elif trend1==trend2 and trend1 in [4,8]:
    if alfa1>alfa2:
        direction="left"
    if alfa1<alfa2:
        direction="right"
elif (trend1<trend2) or (trend1==8 and trend2==1) or (trend1==7 and trend2==1) or
(trend1==8 and trend2==2):
    if (trend1==1 and trend2==8) or (trend1==1 and trend2==7) or (trend1==2 and trend2==8):
        direction="left"
    else:
        direction="left"
elif trend1>trend2 or (trend1==1 and trend2==8) or (trend1==1 and trend2==7) or (trend1==2
and trend2==8):
    direction="left"
else:
    direction="left"

```

Po výpočtu úhlové změny následuje určení její orientace „direction“. Ta nabývá hodnot „right“, „left“ a „0“. Nástroj opět, jako v předešlém případě, využívá soustavu podmiňovacích příkazů. Určení orientace vychází z hodnoty „trend“. Např. na Obr. 9 je v případě segmentů 2 a 3 orientace pravá, což odpovídá směru otáčení hodinových ručiček.

```

if beta < 0.1:
    beta=0
    direction="0"

```

Neboť je nástroj velmi přesný a počítá v řádech desítek desetinných míst, je nutné stanovit nejnižší hranici pro veškeré úhly, které nástroj vypočte. Hranice je stanovena na  $0,1^\circ$ . Úhly menší nástroj vyhodnotí jako nulové.



```

row.UhelPlus = beta
row.DirectionPlus = direction
cur.updateRow(row)

```

Výsledné hodnoty jsou zapsány do pole „UhelPlus“ a „DirectionPlus“. Protože se jedná o poslední editaci v tomto řádku, je nutné provést operaci „save edits“, která je zastoupena příkazem „updateRow“.

```

startx1 = startx2
starty1 = starty2
endx1 = endx2
endy1 = endy2

```

Před přesunem kurzoru na další řádek je nutné přepsat souřadnice prvního segmentu. Segment, který byl v tomto výpočtu načítán jako druhý (FID+1) je pro další řádek, tedy další kolo výpočtu označen jako první (FID).

```

del cur, row

```

Po skončení všech opakování, tedy v okamžiku, kdy kurzor dosáhne posledního řádku, je skript ukončen příkazem, který načtený kurzor a číslo řádku odstraní.

## 4.2. Skript 2 – Úprava koncových úhlů

Skript upravuje hodnoty úhlů u koncových segmentů linií. Řeší tak problém výpočtu úhlů u nenavazujících segmentů jak je popsán v kapitole 6.2.

```

import arcpy, math
originalinputfeature = arcpy.GetParameterAsText(0)
outputfeature = arcpy.GetParameterAsText(1)
arcpy.CopyFeatures_management(originalinputfeature, outputfeature)

```

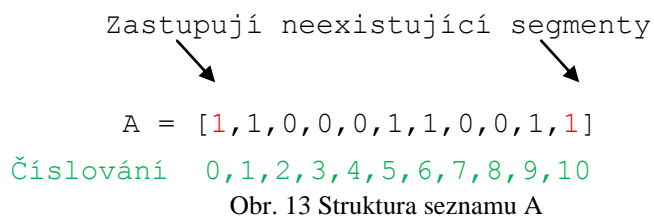
V úvodu skriptu se nacházejí základní příkazy popsané v kapitole 4.1.

```

cur = arcpy.SearchCursor(outputfeature)
A = [1]
for row in cur:
    if row.END == "end" or row.END == "crossroad":
        A.append(1)
    else:
        A.append(0)
del cur, row
A.append(1)

```

V prvním kroku je vytvořen seznam s názvem „A“ který je nositelem informace o tom, který segment obsahuje atribut „end“ nebo „crossroad“. Takový segment je v seznamu označen číslem 1. Ostatní segmenty jsou zastoupeny číslem 0. Struktura tohoto seznamu je zobrazena na Obr. 13. Aby bylo zabráněno chybám, jsou pro úplnost do seznamu zařazeny i dvě krajní hodnoty „1“, které zastupují neexistující segmenty na koncích linie.



```
cur = arcpy.UpdateCursor (outputfeature)
FID = 1
for row in cur:
    if A[FID]==1:
        if A[FID-1]==1:
            row.UhelMinus = 0
            row.DirectionMinus = "0"
            cur.updateRow(row)
        if A[FID+1]==1:
            row.UhelPlus = 0
            row.DirectionPlus = "0"
            cur.updateRow(row)
    FID = FID + 1
del cur, row
```

V druhém kroku skript načítá prvky seznamu „A“ a podle toho odmazává hodnoty úhlů a orientace u problematických segmentů. Je k tomu použit kurzor pro editaci AT. Hodnota „FID“ odpovídá číslu segmentu. Podle struktury seznamu (Obr. 13) je jasné, že musí být „FID“ před začátkem opakování nastaven na hodnotu 1. Jestliže je segment označen jako problémový (A[FID]=1), skript zkontroluje sousední segmenty. Pokud narazí na hodnotu 1, vymaže hodnotu společného úhlu „beta“ (Obr. 17).

### 4.3. Skript 3 – Řazení segmentů do oblouku

Skript přiřadí každému segmentu identifikační číslo oblouku, ke kterému náleží. Příslušnost k oblouku je dána pravidly popsanými v kapitole 5.2.2.

```

import arcpy, locale, math
locale.setlocale(locale.LC_ALL, '')
originalinputfeature = arcpy.GetParameterAsText(0)
outputfeature = arcpy.GetParameterAsText(1)
arcpy.CopyFeatures_management(originalinputfeature, outputfeature)

```

V úvodu skriptu se nacházejí základní příkazy popsané v kapitole 4.1.

```

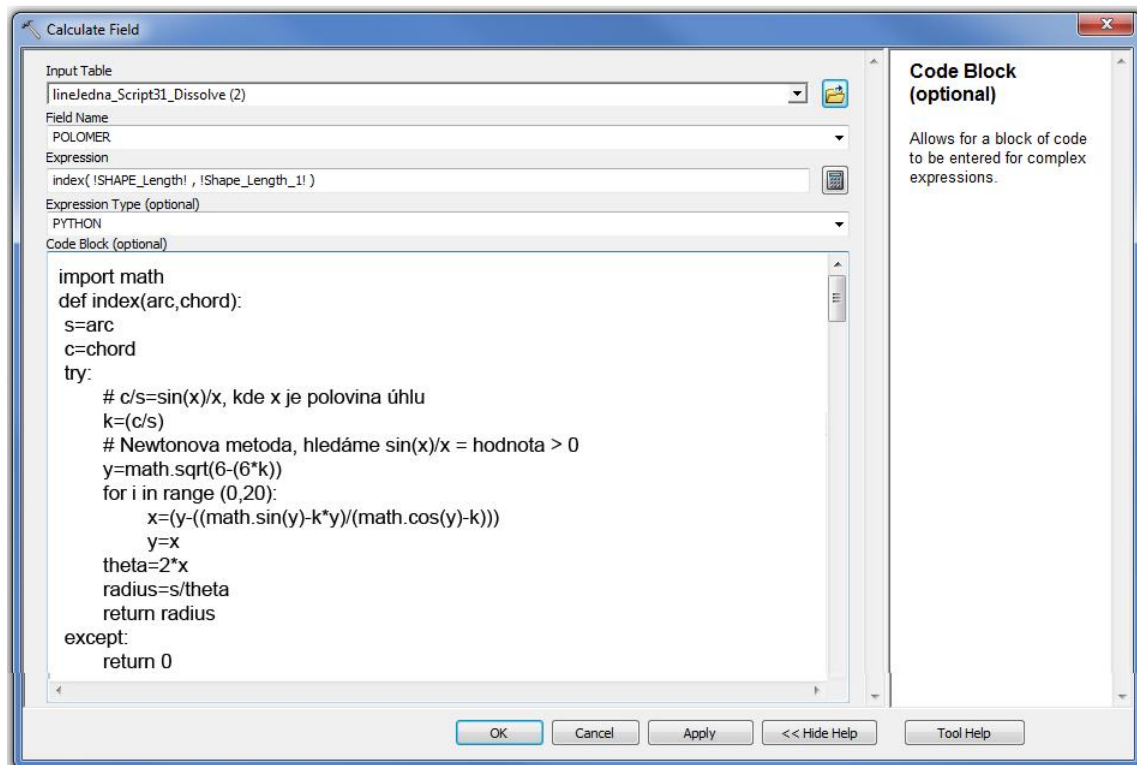
arcpy.AddField_management(outputfeature,"OBLOUK","SHORT","","","","")
cur = arcpy.UpdateCursor (outputfeature)
Curve=0 #hodnota 0 značí, že kurzor se nenachází v oblouku
CurveNumber=1 #identifikátor oblouků
for row in cur:
    if Curve==0: #kurzor se nachází na nulovém úhlu a „čeká na oblouk“
        if row.UhelPlus != 0: #nenulový úhel s následujícím segmentem (nový oblouk)
            Curve=1 #kurzor se ocitl v oblouku
            row.OBLOUK = CurveNumber
            cur.updateRow(row) #zápis identifikátoru
        else:
            Curve=Curve #přesun na další segment beze změny
    else: # kurzor se nachází v oblouku a „čeká na konec oblouku“
        if row.UhelPlus == 0: #nulový úhel s následujícím segmentem (konec oblouku)
            row.OBLOUK = CurveNumber
            cur.updateRow(row) #zápis identifikátoru
            Curve=0
            CurveNumber=CurveNumber+1 #příprava identifikátoru pro nový oblouk
        else:
            row.OBLOUK = CurveNumber
            cur.updateRow(row) #zápis identifikátoru
del cur, row

```

Do AT je vloženo nové pole „OBLOUK“ ve formátu „SHORT“ pro zápis identifikátoru oblouku „CurveNumber“. Poté kurzor prochází postupně záznamy v AT (segmenty linie) a hledá nenulový úhel. Pokud na něj narazí, identifikuje začátek oblouku. Do té doby než narazí na úhel nulový, vepisuje do pole „OBLOUK“ totožnou hodnotu identifikátoru. V případě posledního segmentu s nenulovým úhlem zvýší hodnotu identifikátoru o 1 pro oblouk, který bude následovat. Výsledkem je AT obsahující pole „OBLOUK“ kde jsou pro jednotlivé záznamy uvedeny identifikátory oblouků (ukázka na Obr. 23).

#### 4.4. Newtonova metoda pro Field Calculator

Nástroj „Field Calculator“ (v „Model Builder“ nazývaný „Calculate Field“) je přizpůsoben k provádění výpočtů v rámci jednotlivých záznamů v AT. Umožňuje nejen zadávání jednoduchých příkazů, ale i možnost použití skriptovacího jazyka pro složitější výpočetní metody. Nástroj byl použit ve třetí části modelu pro výpočet poloměru oblouků (kapitola 5.4.5) Nastavení nástroje je znázorněno na Obr. 14. Do pole „Expression“ je zadána funkce s názvem „index“, která načítá hodnoty polí „SHAPE\_Length“ (délka oblouku) a „Shape\_Length\_1“ (vzdálenost koncových bodů oblouku). Ve volitelném poli „Code Block“ je pak definována funkce „index“ za pomoci skriptu v jazyce Python. Pro provedení výpočtu je nutné importovat modul „math“, který obsahuje příkazy např. pro použití trigonometrických funkcí. Tělo skriptu za funkcí „try“ odpovídá Newtonově metodě popsané v kapitole 3.2.1. Příkaz „return“ vrací do pole „Expression“ hodnotu „radius“. Pokud se nepodaří hodnotu nalézt, vrátí příkaz nulovou hodnotu.



Obr. 14 Field Calculator, výpočet poloměru oblouku

## **5. MODEL PRO HODNOCENÍ HORIZONTÁLNÍHO ZAKŘIVENÍ**

### **5.1. Vstupní data**

Pro sestavování funkčního modelu byl brán ohled na strukturu vstupních dat. Jelikož se data od různých poskytovatelů strukturálně liší, byla vybrána data, která běžně využívá Magistrát města Liberce, Úřad územního plánování Liberec (ÚÚP) a např. Ředitelství hasičského záchranného sboru Libereckého kraje. Jedná se o výstupy z územně analytických podkladů (ÚAP), jevy A089 až A093. Sledované jevy ÚAP jsou exportovány z databáze ÚÚP Liberec, která přejímá geometrickou složku dat ze ZABAGED. V předávacím protokolu vydaném Ing. Markem Veselým dne 24.10.2011 je zmíněno: „Polohové a geometrické zobrazení dat je shodné s daty od poskytovatelů údajů. Kvalita a aktuálnost dat vydaných údajů od ÚÚP je přímo úměrná kvalitě a aktuálnosti poskytnutých údajů od jednotlivých poskytovatelů.“ Lze tedy předpokládat, že model bude funkční pro veškerá data získaných ze ZABAGED.

Geometrickým základem použitých dat je objekt AP001 ZABAGED, kde geometrické určení objektu tvoří linie, tedy osa komunikace s geometrickou přesností B. To znamená, že střední polohová chyba dosahuje hodnoty 5 metrů. Původním zdrojem geometrie je ZM10, SM50, letecké měřické snímky, ortofoto a šetření v terénu.

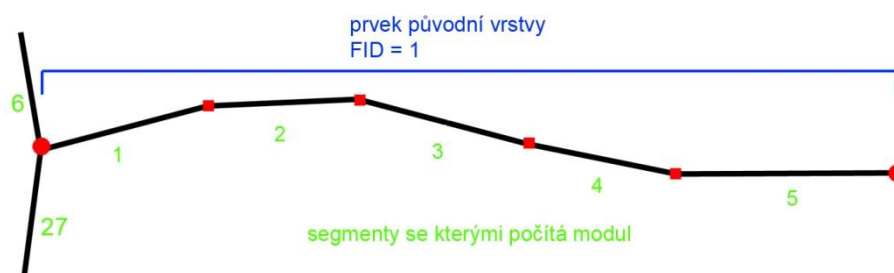
Poskytnutá data jsou ve vektorové reprezentaci, kde každý prvek odpovídá úseku komunikace. Tento úsek je ohraničen buď křižovatkou, či hranicí území. Prvky mimo jiné obsahují velké množství atributů, jako např. číslo úseku, třídu silnice, datum zaznamenání. Pro potřeby modelu jsou významné pouze atributy FID (identifikátor), Shape (geometrie) a ShapeLenght (délka prvku), které jsou v průběhu analýzy modifikovány. Ostatní atributy jsou zachovány v původní podobě.

### **5.2. Teoretické předpoklady modelu**

#### **5.2.1. Segmentace linie**

Základním prvkem, se kterým model počítá, je úsečka, v modelu nazývaná segment. Je proto nutné původní prvky rozdělit tak, aby každý segment měl jedinečný identifikátor, jak

ukazuje Obr. 15. Identifikátor (FID) pak určuje návaznost segmentů. Předpokladem pro model tedy je, že sousední segmenty mají vždy rozdíl atributů FID roven jedné (s výjimkou míst křížení). Neplatí však, že segmenty s rozdílem FID rovným jedné jsou nutně sousedící (více v kapitole 5.2.5). Nositelem atributu jsou tedy ve výsledku jednotlivé segmenty. Po linii se však dá pohybovat dvěma směry, proto je třeba, aby každý segment nesl informaci jak o úhlové změně se segmentem předchozím (FID -1), tak s následujícím (FID+1). Dva segmenty nesou ve výsledku údaj o hodnotě společné úhlové změny (Obr. 16)



Obr. 15 Rozdělení úseku na segmenty

### 5.2.2. Přiřazení segmentu k oblouku

Po výpočtu úhlové změny pro jednotlivé segmenty se tyto spojí do oblouků, pro které bude následně vypočten poloměr. Po řazení segmentů do oblouku platí:

1. Segment nenáleží žádnému, nebo právě jednomu oblouku.
2. Po sobě jdoucí segmenty se stejnou orientací nenulového úhlu náležejí jednomu oblouku
3. Hranici dvou oblouků tvoří alespoň jeden nulový úhel.

### 5.2.3. Segmenty s opačnou orientací úhlu

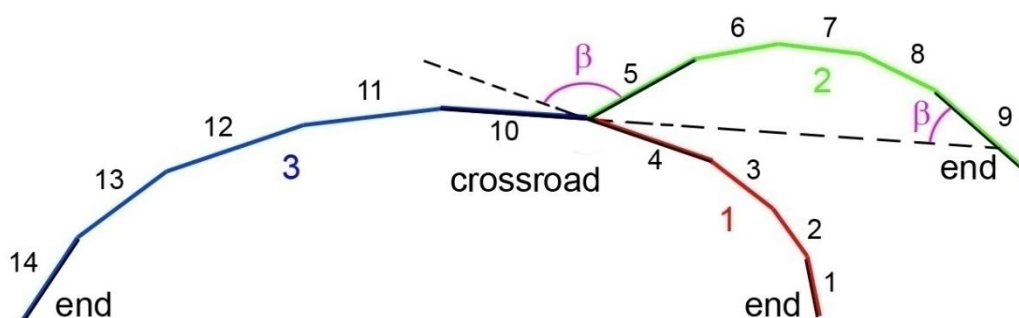
Předpokladem modelu je, že při rozdělení linie komunikace na jednotlivé oblouky bude každý segment náležet právě jednomu oblouku. Problém nastává v případě segmentu s opačnou orientací úhlů, jako je např. segment číslo 3 na Obr. 15. Jak vyplívá z obrázku, náležel by jak kladnému oblouku tvořenému segmenty 1-2-3, tak navazujícímu zápornému oblouku 3-4-5. Pro správné určení hranice dvou oblouků (podle 6.2.2) je nutné segment 3 rozdělit na dvě rovnoběžné, stejně dlouhé části s nulovým společným úhlem. Každá z těchto částí je pak přiřazena jednomu oblouku. Tato oprava je provedena v první části modelu popsané v kapitole 5.4.1. Dva segmenty (záznamy v AT) na Obr. 16. se stejnou délkou, nulovým společným úhlem a opačnou orientací navazujících úhlů jsou příkladem rozděleného původního problematického segmentu.

OBJECTID *	END	UhelSegment-1	UhelSegment+1	DirectionMinus	DirectionPlus	Shape Length
518 <Null>		1,322576	0	left	0	7,426195
519 <Null>		0	0,827683	0	right	7,426195
520 <Null>		0,827683	1,874135	right	right	101,01864
521 <Null>		1,874135	0	right	0	67,124018
522 <Null>		0	4,143793	0	left	67,124018
523 <Null>		4,143793	0	left	0	5,336193
524 <Null>		0	0,546529	0	right	5,336193
525 <Null>		0,546529	0	0	0	4,215392
526 <Null>		0	5,143934	0	left	5,143934
527 <Null>		5,143934	0	0	0	5,749781
528 <Null>		0	3,977837	0	right	5,749781
529 <Null>		3,977837	2,332802	right	right	28,315878
530 <Null>		2,332802	47,66354	right	right	25,962947
531 <Null>		47,66354	0	right	0	114,541595
532 <Null>		0	24,396867	0	left	114,541595
533 <Null>		24,396867	24,597507	left	left	12,956458
534 <Null>		24,597507	0,397671	0	left	16,329864
535 <Null>		0,397671	0	0	0	8,861115
536 <Null>		0	25,047225	0	right	8,861115
537 <Null>		25,047225	0	right	0	4,281442
538 <Null>		0	28,366687	0	left	4,281442
539 crossroad		28,366687	0	left	0	92,959826
540 crossroad		0	15,598392	0	left	11,887725
541 <Null>		15,598392	4,590992	left	left	36,917464
542 <Null>		4,590992	39,122536	left	left	9,433427
543 <Null>		39,122536	0,228977	left	left	6,725504
544 <Null>		45,917383	0,228977	left	left	7,623687

Obr. 16 Ukázka AT po dokončení druhé části modulu

#### 5.2.4. Křížení linií

Problém, který je třeba vyřešit pro správnou funkčnost modelu, představuje mimo jiné také křížení linií neboli křižovatky komunikací. Nastane-li situace, kdy se do zatáčky na hlavní silnici připojuje silnice vedlejší, bylo by vhodné, aby se z pohledu jízdy po hlavní silnici jednalo pouze o zatáčku. Na rozdíl od vedlejší silnice, pro kterou se jedná o křižovatku. Setká-li se spolu více segmentů v rámci jedné vrstvy dat (Obr. 17), nelze jednoznačně určit, která linie představuje silnici hlavní a která vedlejší. Model proto nemá možnost určit mezi kterými segmenty provést výpočet úhlu a následné přiřazení k oblouku. Jak vyplývá z Obr. 17, úseky na sebe navazují podle toho, v jakém pořadí byly digitalizovány. Bez provedení úpravy by nástroj, který se řídí pouze podle FID, vypočítal úhlovou změnu pro segmenty 4-5 a zahrnul tak segment 5 do oblouku 2. Problém je vyřešen tím, že u segmentů obsahující atribut „crossroad“ (segment se dotýká bodu křížení) jsou anulovány společné úhly. Tato operace je součástí první části modelu a je popsána v kapitolách 4.2 a 5.4.2. Výsledkem je nulová úhlová změna pro segmenty 4 a 5, která se stává dělicím bodem pro oblouky 1 a 2.



Obr. 17 Problematické úhly na koncích linií a při křížení

Je-li k dispozici údaj o třídě komunikace, je možné v místech křížení správně určit jak zatáčku, tak křižovatku, ovšem pouze při předpokladu, že při styku dvou rozdílných tříd bude jako hlavní vždy komunikace s vyšší třídou. Nástroj lze použít pro jednotlivé třídy zvlášť. Následnou identifikací bodů křížení (např. nástrojem „Intersect“) mezi jednotlivými vrstvami, lze dosáhnout požadovaného výsledku.

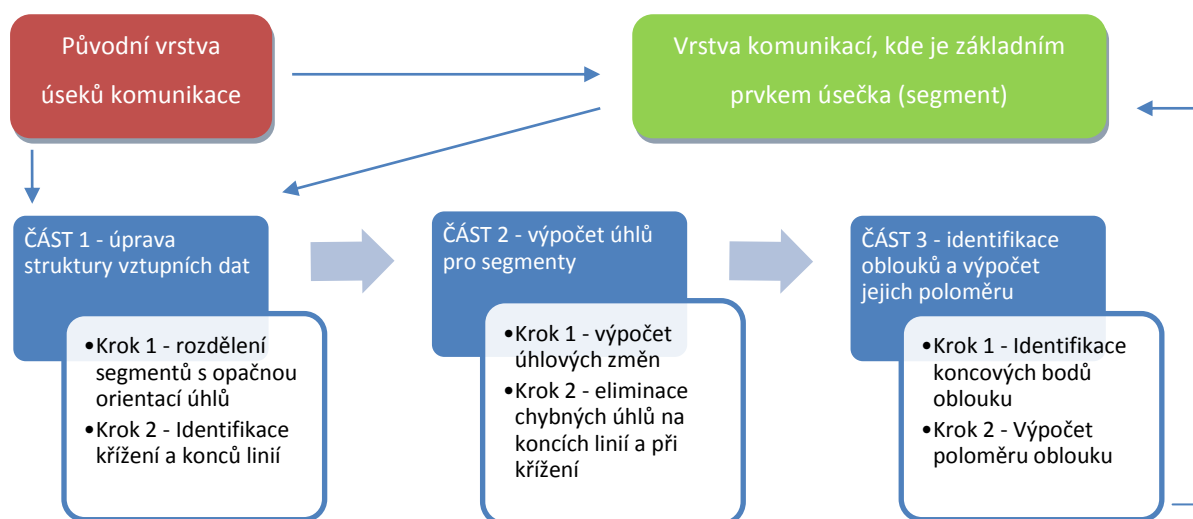


### 5.2.5. Koncové segmenty linií

Dalším problémem představují konce linií. Jednoduše řečeno segmenty, které mají pouze jednoho souseda. Podstatou problému je skutečnost, že atribut FID všech prvků ve vrstvě, je generován z nepřerušované číselné řady. Vzdálené segmenty, jako jsou 5 a 6 na Obr. 17, proto mohou obsahovat sousední čísla v atributu FID. Proto je nutné konce původních linií definovat speciálním atributem „end“, který určí, je-li segment koncový. Oprava poté probíhá odstraněním společného úhlu, tedy stejným způsobem jako v případě křížení (kapitola 5.2.4).

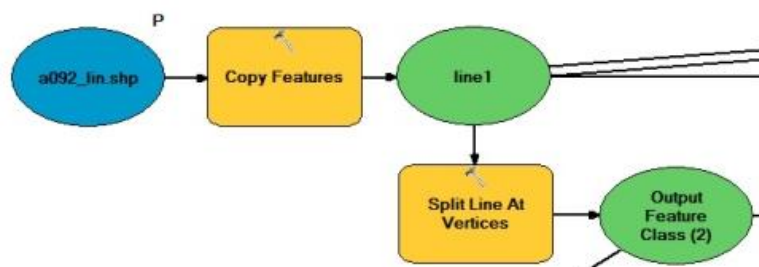
## 5.3. Struktura modulu

Zjednodušená struktura celého modelu je znázorněna na Obr. 18. Model je rozdělen do tří částí, kterým předchází úprava původní linie, jak je popsáno v kapitole 5.2.1. Tím získá vrstva strukturu, do které budou zapsány výsledné atributy. Celý model je navržen v prostředí „Model Builder“, kde jednomu kroku odpovídá několik vzájemně navázaných operací. Podrobná struktura modelu je popsána v následujících kapitolách. Pro lepší orientaci ve struktuře modelu, je vždy v zobrazovaných krocích zakreslena i předcházející operace.



Obr. 18 Struktura modelu

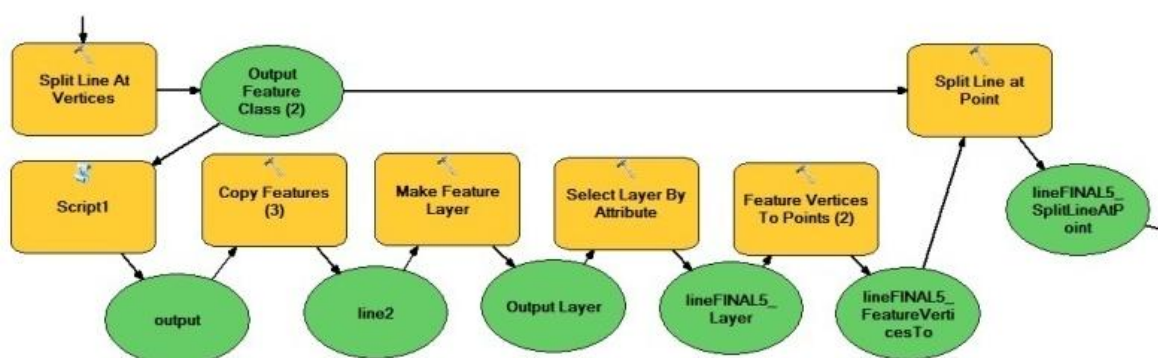
## 5.4. Vstup



Obr. 19 Vstupní část modelu

Jako vstupní parametr je nastaven „Feature Class“ ve formátu „line“, nebo „polyene“. Operaci „Copy Features“ je zajištěno, že původní vrstva nebude přepsána. Cesta pro uložení nového souboru je nastavena do základní geodatabáze ArcGIS ve které také probíhají veškeré další operace. To je vhodné pro hladký průběh všech procesů. Kopírováním vzniká vrstva „line1“ (veškeré názvy vrstev jsou vytvořeny autorem pouze pro potřeby modulu). Následuje rozdělení funkcí „Split Line At Vertices“, která rozdělí linii podle tzv. vertexů neboli vrcholů. Takto vzniklé vrstvy vstupují do dalších částí modelu, jak naznačují neukončené čáry v Obr. 19.

### 5.4.1. Část 1 - rozdělení segmentů s opačnou orientací úhlů



Obr. 20 Struktura modelu, rozdělení segmentů

V tomto kroku je řešen problém segmentů s opačnou orientací úhlů (popsáno v kapitole 5.2.3). Segmentovaná vrstva zde vstupuje do nástroje „Split Line At Points“, který rozdělí problematické segmenty. Bodová vrstva pro tento nástroj je vytvořena v pěti krocích ve spodní části Obr. 20. V prvním kroku je aplikován autorem navržený nástroj zmíněný

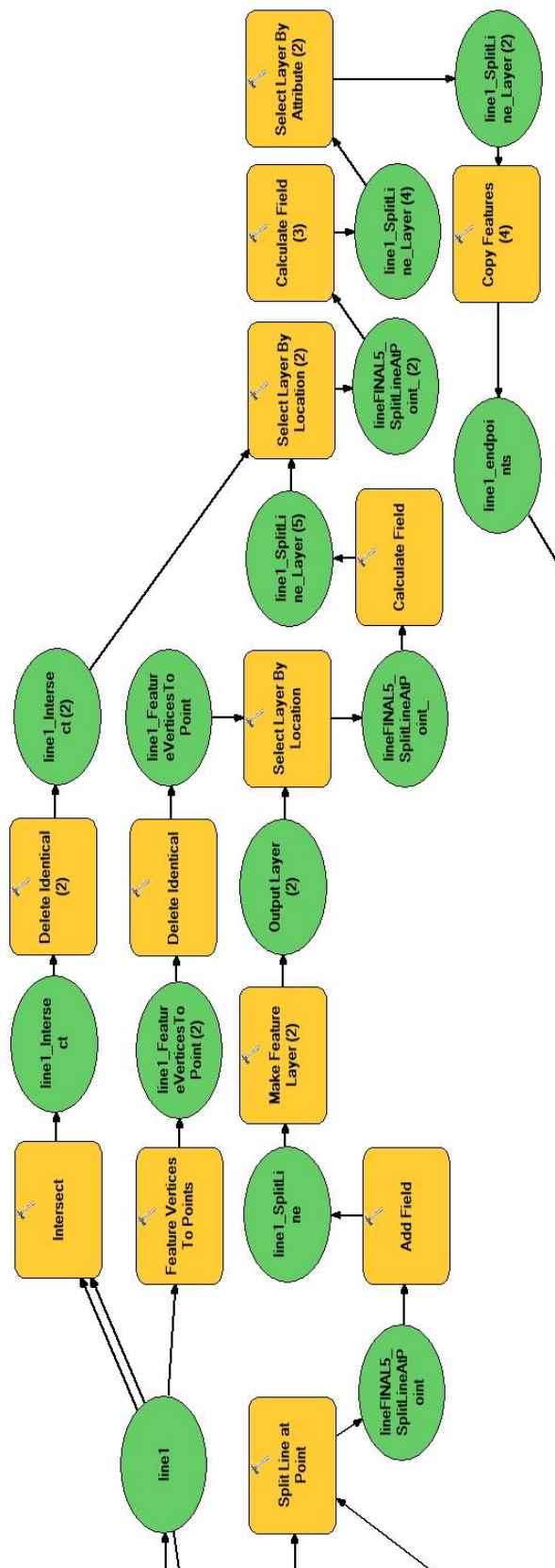
v kapitole 4.1. Ten přidá každému segmentu linie atribut o orientaci úhlů, který svírá se sousedními segmenty. Po operaci kopírování, která slouží ke stabilizaci souvislého běhu modelu, jsou pomocí funkce „Select Layer By Attribute“, která je shodná s funkcí „Select By Attribute“ jen je provedena v rámci vytvořené vrstvy („Feature Layer“), označeny segmenty obsahující rozdílné atributy v polích „DirectionPlus“ a „DirectionMinus“. Příkaz v SQL pro tuto operaci je:

```
("DirectionMinus" = 'left' AND "DirectionPlus" = 'right') OR ( "DirectionMinus" = 'right' AND "DirectionPlus" = 'left' )
```

Vprostřed každého vybraného segmentu je pomocí funkce „Feature Vertices To Points“ s nastavením „MID\_POINT“ vytvořen bod. Vzniklá bodová vrstva je použita pro rozdělení segmentů v původní linii nástrojem „Split Line At Points“

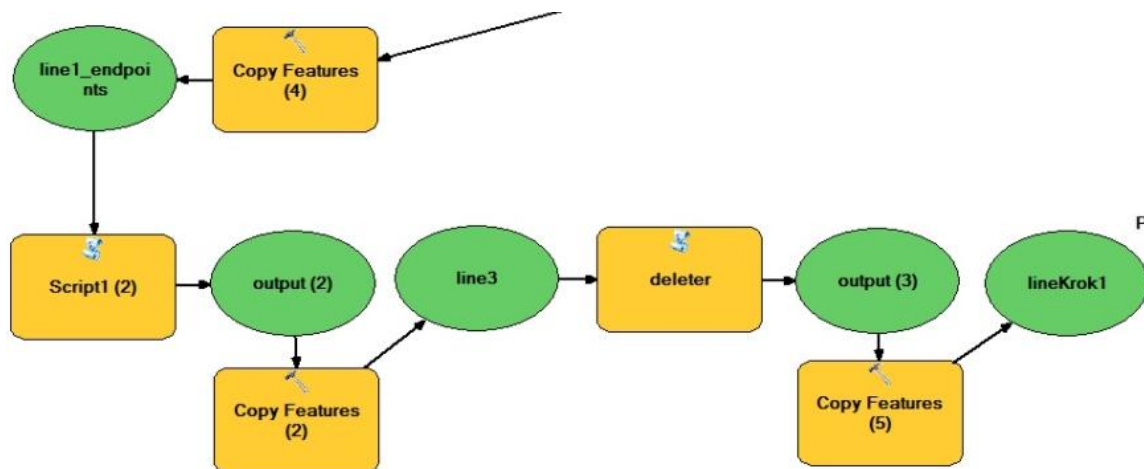
#### **5.4.2. Část 1 – Identifikace křížení a konců linií**

V tomto kroku je vyřešen problém popsáný v kapitole 5.2. Pro koncové segmenty úseků jsou vepsány atributy „crossroad“ pro křížení a „end“ pro jiná ukončení (zpravidla se jedná o linie, které byly přerušeny při tzv. ořezávání dat např. podle hranice území). Nejprve jsou vytvořeny dvě bodové vrstvy. První vzniká pomocí nástroje „Feature Vertices to Point“ s nastavením „BOTH\_ENDS“ nad původní nesegmentovanou vrstvou. Vytvořenými body jsou označena místa konců linií. Druhá pak vzniká také z původní vrstvy obsahující celé úseky komunikací prostřednictvím nástroje „Intersect“ a označuje místa křížení. Mezitím je v cílové vrstvě vytvořeno nové pole „END“ textového typu pro zápis nových atributů. Nástroj „Select Layer By Location“ s nastavením „INTERSECT“ označí segmenty dotýkající se bodové vrstvy konců linií a prostřednictvím „Calculate Field“ je pro tyto segmenty vepsán atribut „end“. Následuje stejný postup pro body křížení. U těch segmentů, které jsou ukončeny křížením je atribut „end“ přepsán na „crossroad“. Následuje „Select By Attribute“ nastavené na „Clear\_Selection“ pro zrušení označení a kopírování do nové vrstvy, která vstupuje do druhé části.



Obr. 21 Striktura modelu, křížení a konce linií

### 5.4.3. Část 2 - výpočet úhlových změn v průběhu linie



Obr. 22 Struktura modelu, úhlová změna

Pro výpočet úhlové změny segmentů upravené linie je opět použit Skript 1 popsáný v kapitole 4.1. Pro anulování společných úhlů problematických segmentů (kapitoly 5.2.4 a 5.2.5) je aplikován Skript 2 (daleter) popsáný v kapitole 4.2. Výstupem je vrstva „Feature Class“ s volitelným úložištěm. Uživatel v této chvíli může vrstvu použít pro vlastní analýzy, nebo přistoupit k třetí části modulu.

### 5.4.4. Část 3 - Identifikace koncových bodů oblouku

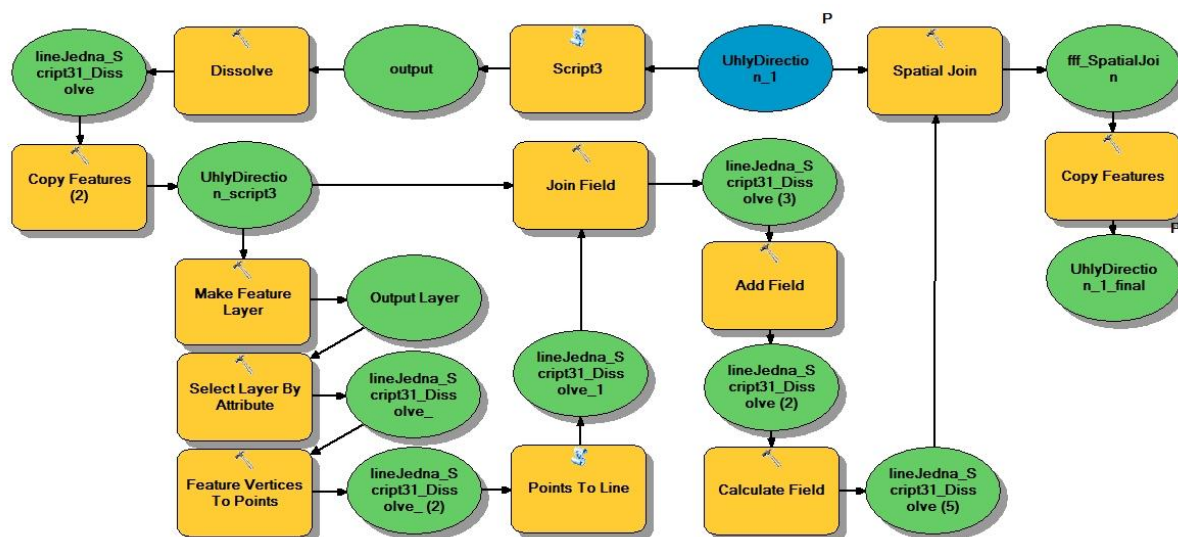
V této části modelu je aplikován Skript 3 popsáný v kapitole 4.3. Výsledná AT je znázorněna na Obr. 23, kde je vyznačen jeden z identifikovaných oblouků. Prvním segmentem zařazeným do oblouku je vždy segment s nulovou hodnotou atributu „UhelMinus“. V průběhu oblouku mají úhlové změny vždy stejnou orientaci. Nulová hodnota atributu „UhelPlus“ značí poslední segment daného oblouku.

TARGET_FID	END	UhelMinus	UhelPlus	DirectionMinus	DirectionPlus	OBOUK	POLOMER	Shape_Length
536		0	25,047224733598	0	right	147	18,5101462524655	8,86111546306745
537		25,047224733598	0	right	0	147	18,5101462524655	4,28144160888233
538		0	28,3666868626163	0	left	148	278,641087692608	4,28144160879311
539	crossroad	28,3666868626163	0	left	0	148	278,641087692608	92,959825668169
540	crossroad	0	15,5983915714986	0	left	149	35,9572099952097	11,8877253080957
541		15,5983915714986	4,59099242510409	left	left	149	35,9572099952097	36,9174641734252
542		4,59099242510409	39,1225355393413	left	left	149	35,9572099952097	9,43342700193397
543		39,1225355393413	45,9173828333776	left	left	149	35,9572099952097	6,72550369861033
544		45,9173828333776	0,22897723339036	left	left	149	35,9572099952097	7,6236865097559
545		0,22897723339036	0	left	0	149	35,9572099952097	12,247044061699
546		0	11,8484713592111	0	right	150	63,5302592999159	12,2470440618556
547		11,8484713592111	0	right	0	150	63,5302592999159	10,4440101612333
548		0	2,34398646528704	0	left	151	1263,58661707989	10,444010161397
549		2,34398646528704	0,603066870580371	left	left	151	1263,58661707989	47,3878643114703

Obr. 23 Ukázka AT po aplikaci Skriptu 3

### 5.4.5. Část 3 - Výpočet poloměru oblouku

Po získání atributu s identifikátorem oblouků pomocí Skriptu 3 přejde model k výpočtu hodnoty poloměru pro každý oblouk. Nástrojem „Dissolve“ je provedeno sloučení prvků na základě atributu „OBLOUK“. Ze vzniklé vrstvy jsou nástrojem „Select Layer By Attribute“ vybrány pouze prvky zastupující nějaký oblouk (*SQL: "OBLOUK" IS NOT NULL*). Nástroj „Feature Vertices To Points“ s nastavením „BOTH\_ENDS“ vytvoří bodovou vrstvu tvořenou koncovými body těchto prvků. Pomocí nástroje „Points To Line“ kde pro volbu „Line Field“ je zvolen atribut „OBLOUK“ jsou tyto body spojeny do úseček. FID těchto úseček odpovídá FID oblouků ve sloučené vrstvě a jejich délka je rovna vzdálenosti koncových bodů těchto oblouků. Nastává situace znázorněná na Obr. 3. Na základě atributu „OBLOUK“ je atribut „Shape Length“ z vrstvy úseček přidán do vrstvy oblouků nástrojem „Join Field“.



Obr. 24 Struktura modelu, výpočet poloměru oblouků

Nacházejí-li se ve vrstvě sloučených oblouků atributy s hodnotami jak pro délku oblouků, tak pro vzdálenost jejich koncových bodů, lze přistoupit k výpočtu poloměru za pomoci Newtonovy metody (popsána v kapitole 3.2.1). Výpočet je proveden nástrojem „Field Calculator“ (kapitola 4.4) a výsledky jsou zapsány do nového pole „POLOMER“ ve formátu „DOUBLE“. Získané hodnoty poloměru oblouků jsou vepsány do původní (vstupní) segmentované vrstvy a to nástrojem „Spatial Join“ nastavené na „CONTAINS“. Všechny segmenty, ze kterých byl v úvodu tohoto kroku oblouk sestaven, přebírají hodnotu poloměru oblouku. Výsledkem je segmentovaná vrstva komunikací, kde každý segment nese údaj jak o

úhlové změně, tak o poloměru oblouku ve kterém se nachází (ukázka AT na Obr. 23). Pro výstup je nastaveno volitelné úložiště.

## 6. POTVRZENÍ FUNKČNOSTI MODELU

Pro potvrzení funkčnosti navrženého modelu byla provedena síťová analýza nad vektorovými daty silniční sítě pro území ORP Liberec popsané v dalších částech této kapitoly. Použitými daty byly jevy ÚAP Libereckého kraje:

- A089 Rychlostní silnice
- A090 Silnice I. třídy
- A091 Silnice II. třídy
- A092 Silnice III. třídy

Cílem analýzy bylo porovnat výsledky před a po použití modelu a stanovit, zdali je model použitelný pro zdokonalení síťových analýz v oblasti horizontálního zakřivení komunikací.

### 6.1. Analýza průběhu komunikace

#### 6.1.1. Křivolakost

Výsledný datový model obsahuje údaje o úhlových změnách jednotlivých segmentů komunikace. Díky tomu je možné vypočítat hodnotu křivolakosti pro libovolné úseky silniční sítě. Výpočtem této hodnoty pro jednotlivé typy silnic v rámci území ORP Liberec byly získány výsledky prezentované v Tab. 6. Z výsledků je jasně patrné, že hodnota křivolakosti stoupá pro nižší třídy silnic. To odpovídá předpokladu, že pro stavbu vyšších tříd silnic jsou kladeny vysoké požadavky na snížení křivolakosti. Zároveň tyto výsledky nepřímo potvrzují funkčnost navrženého datového modelu.

Tab. 6 Křivolakost jednotlivých tříd silnic na území ORP Liberec

třída komunikace	délka (m)	součet úhlových změn (stupně)	křivolakost
rychlostní silnice	40094,3	2755,9	76
silnice I. třídy	98312,6	17075,6	193
silnice II. třídy	73375,5	14105,1	214
silnice III. třídy	324018	90134,8	309



## 6.2. Rizikové úseky

Jedna z možností, jak využít navržený datový model pro analýzu rizikových úseků v průběhu komunikace, je určení bodů s vysokou úhlovou změnou. Výběrem hodnot z AT dané komunikace lze jednoduše identifikovat segmenty s definovanými hodnotami úhlových změn. V příkladu na Obr. 25 jsou identifikovány úhlové změny větší než  $20^\circ$  v části silnice třetí třídy na území obce Liberec. Společně s dalšími vhodně zvolenými postupy by bylo možné vytvořit model pro identifikaci rizikových míst. Zmíněnými dalšími postupy by mohlo být např. vložení atributu s informací o vertikálním zakřivení, o výskytu staveb, vegetace a vodních ploch v blízkosti komunikace, o intenzitě provozu v daném úseku, o výskytu křižovatek, železničních přejezdů, mostů a dalších faktorů ovlivňujících riziko průjezdu.



Obr. 25 Identifikované úhlové změny v průběhu linie



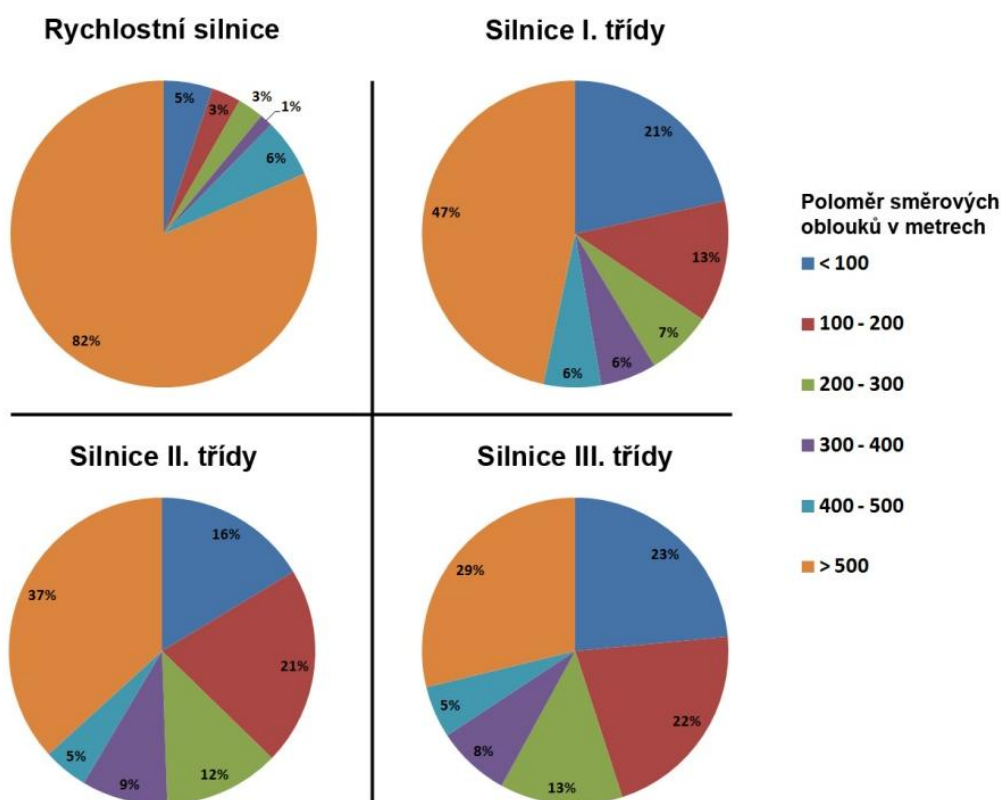
### 6.2.1. Poloměr oblouků

Pro identifikaci rizikových míst i pro určení rychlosti průjezdu lze použít také atribut poloměru směrových oblouků, který je také obsahem modelu. Na Obr. 26 je znázorněna tatáž část komunikace, jako v předchozím případě, rozdělená na jednotlivé oblouky s hodnotami jejich poloměrů. Pro větší názornost jsou oblouky odlišeny barevně podle velikosti. Červená barva v tomto případě upozorňuje na oblouky s poloměrem menším než 100 metrů. Směrový oblouk s nejmenším poloměrem (48m) byl lokalizován v levé části obrázku, což se neshoduje s největší úhlovou změnou z Obr. 25. Z toho vyplývá, že metoda určování poloměru oblouků vyžaduje jisté zdokonalení, aby mohla být prakticky využitelná.



Obr. 26 Poloměry oblouků na vybraném úseku silnice

Výpočet poloměru oblouků byl (jako v případě křivolakosti) proveden pro celé území ORP Liberec. Hodnoty poloměrů byly rozděleny do šesti skupin podle velikosti a to pro každou třídu komunikace. Výsledky jsou demonstrovány čtveřicí grafů na Obr. 27. Z výsledků vyplývá, že 82% všech směrových oblouků na rychlostních silnicích má poloměr větší než 500 metrů. Zbylé oblouky odpovídají výrazně zakřiveným nájezdům a sjezdům, které jsou součástí vektorové vrstvy (A089). Pro zbylé třídy silnic klesá procento oblouků s velkým poloměrem úměrně k růstu křivolakosti. Rozdíl v zastoupení jednotlivých poloměrů mezi třídami komunikace odpovídá výsledkům v Tab. 6.



Obr. 27 Zastoupení poloměrů oblouků na území ORP Liberec pro jednotlivé třídy silnic

### 6.3. Sít'ová analýza

Funkčnost navrženého datového modelu byla ověřena pomocí sít'ové analýzy časové dostupnosti v rámci území OPR Liberec. Bod, ke kterému se časy vztahují, byl umístěn v blízkosti výjezdního místa HZS Libereckého kraje v Liberci v Šumavské ulici. Analýza byla provedena za použití tří odlišných postupů pro určení rychlosti vozidla při pohybu po síti. Postupy byly zvoleny tak, aby na výstupech analýzy bylo možné ověřit funkci navrženého datového modelu. Každý postup je reprezentován jedním grafickým výstupem. Pro výstup č. 1 (Příloha 4) byla rychlost odvozena výhradně od třídy komunikace, tedy bez využití navrženého modelu. Určení rychlosti pro výstup č. 2 (Příloha 5) vychází navíc z hodnoty křivolakosti. Rychlost vozidla ve výstupu č. 3 (Příloha 6) je upravena v závislosti na poloměru směrových oblouků. Postupy určení rychlostí pro všechny tři výstupy jsou uvedeny v kapitole 6.3.1.

### 6.3.1. Určení rychlosti průjezdu komunikací

Průměrná rychlost pro všechny tři analýzy je odvozena od třídy komunikace. Tab. 8 obsahuje údaje o stanovené rychlosti. Hodnoty vycházejí z maximální povolené rychlosti a jsou upraveny za předpokladu výskytu faktorů snižující rychlost, jako jsou např. hustota provozu, omezení rychlosti v obcích a podobně.

Tab. 8 Určení průměrné rychlosti

třída komunikace	průměrná rychlost (km/h)
rychlostní silnice	110
silnice I. třídy	80
silnice II. třídy	70
silnice III. třídy	60

Tab. 7 Úprava rychlosti podle křivolakosti

křivolakost	průměrná rychlost
menší než 100	100%
100 až 200	90%
201 až 300	80%
301 až 400	70%
více než 400	60%

Úprava průměrné rychlosti pro výstup č. 2 byla provedena na základě křivolakosti úseků. Ta byla vypočtena ze součtu úhlových změn a délky úseků podle vzorce uvedeného v kapitole 3.1. Hodnota původní rychlosti byla následně redukována procentuelně jak udává Tab. 7.

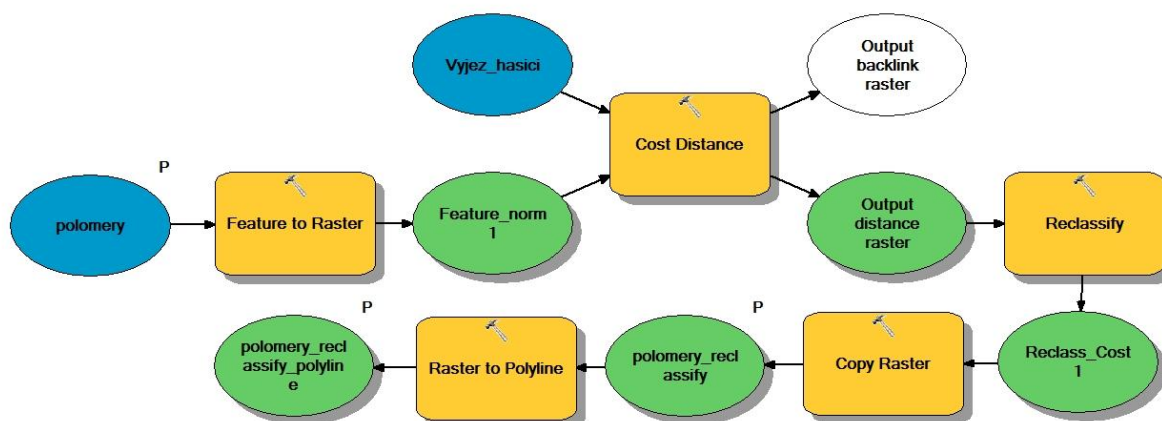
Tab. 9 Doporučené poloměry směrových oblouků v závislosti na rychlosti a dostředném sklonu (ČSN 2004)

Návrhová/ směrová rychlost v km/h	Poloměr kružnicového oblouku v metrech										
	při dostředném sklonu vozovky v %										se základním příčným sklonem 2,5 % <sup>7)</sup>
	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	
130	2450	2050	1750	1525	1350	1225	1125	1025	-	-	4500
120	2075	1750	1500	1300	1150	1050	950	850	-	-	3800
110	1750	1450	1250	1100	925	825	800	725	-	-	3200
100	1450	1200	1050	900	800	720	650	600	-	-	2700
90	1200	1000	850	750	650	600	550	500	-	-	2200
80	775	650	550	500	450	400	350	325	-	-	1700
70	600	500	425	375	330	300	270	250	-	-	1300
60	450	375	325	270	240	220	200	180	170	-	950
50	300	250	220	190	170	150	140	125	120	110	700
40	200	160	140	120	110	100	90	80	75	70	450
30	110	90	80	70	60	55	50	45	40	35	250

Úprava rychlosti pro výstup č. 3 byla provedena na základě poloměru oblouků. Vodítkem pro určení míry redukce rychlosti byla Tab. 9 převzatá z ČSN (2004) udávající nejmenší dovolené poloměry směrových kružnicových oblouků ve vztahu k uvažované rychlosti a dostřednému sklonu. Údaj o dostředném sklonu vozovky není pro použitá data k dispozici. Je proto uměle stanoven na nejnižší hodnotu 2,5% aby byl rozdíl rychlostí pro jednotlivé hodnoty poloměrů co nejvýraznější. Nejnižší rychlost 30 km/h je stanovena pro oblouky s poloměrem menším než 200 metrů. Hodnota rychlosti dále stoupá podle poloměru, ovšem nikdy nepřekročí původní hodnotu průměrné rychlosti dané třídou komunikace.

### 6.3.2. Použité metody

Analýza byla provedena metodou „Cost Distance“, která je dostupná v rámci nadstavby „Spatial Analyst“. Jedná se o běžně používanou metodu nákladové analýzy pracující nad rastrovými daty. Pro převod výsledků analýzy ze sítě na celou plochu zkoumaného území byla použita metoda „Inverse Distance Weight“. Výsledný rastr byl pro přehlednost reklasifikován po čtyřminutových intervalech. Postup analýzy je znázorněn na Obr. 28.



Obr. 28 Postup tvorby síťové analýzy

### 6.3.3. Porovnání výsledků

Při hodnocení výsledků byl kladen důraz především na vzájemné rozdíly mezi jednotlivými výstupy a ne na konkrétní časové hodnoty. Ty nejsou pro zjištění funkčnosti modelu vypovídající. Rychlosti byly stanoveny uměle pro potřeby porovnání výsledků analýzy a nejsou v souladu s reálným chováním vozidla. Podstatné je zdali se prokáže, že model může být použit pro odlišení rychlosti jízdy pro rovné a zakřivené úseky.

Tab. 10 Výsledky analýzy pro určené body (hodnoty jsou uvedeny v minutách)

bod	třída komunikace (bez modelu)	křivolakost (model)	poloměr oblouků (model)
1. Hodkovice n. M.	12	12	12
2. Jablonné v. P.	24	24	28
3. Český dub	20	20	24
4. Křižany	20	28	28
5. Rudolfov	12	20	20

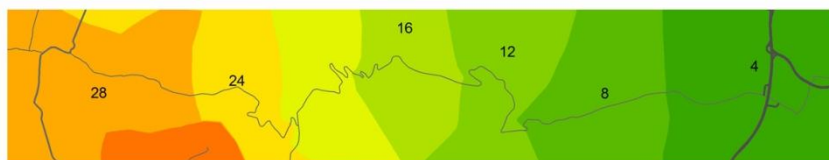
Výstupy analýzy jsou obsaženy v přílohách 4 až 6. Pro jejich porovnání byly na území vytyčeny body (obce) s hodnotou dojezdového času. Hodnoty času jsou uvedeny v Tab. 10. Zkoumány byly rozdíly hodnot mezi jednotlivými výstupy. Pro první bod (Hodkovice n. M.) zůstal čas dojezdu po použití modelu nezměněn. To odpovídá umístění bodu vzhledem k výjezdnímu místu. K dosažení bodu se vozidlo pohybuje převážně po rychlostní silnici R 35 jejíž hodnota křivolakosti i poloměry směrových oblouků nejsou dostatečně velké na to, aby měly vliv na snížení rychlosti. Pro body 2 a 3 byl po použití modelu čas o 4 delší. To značí výraznější zakřivení komunikací při cestě k těmto bodům, což v jistém smyslu odpovídá skutečnosti. V případě bodů, pro jejichž dosažení je nutné překonat velmi výrazné zakřivení (horský terén), byla po použití modelu hodnota času zvýšena až o 8 minut. Výsledkem je potvrzení předpokladu, že lze model použít pro odlišení rychlostí pro úseky s odlišnou mírou zakřivení.

Potvrzení funkčnosti modelu bylo provedeno i porovnáním časových intervalů u konkrétního úseku (Obr. 29). Zkoumaným úsekem je výrazně zakřivená silnice III. třídy vedoucí přes Ještědský hřeben z Liberce do Křižan. Po použití modelu ve výstupech č. 2 a č. 3 se v částech silnice s výrazným zakřivením také podstatně zkrátí interval mezi hodnotami času dojezdu, což odpovídá skutečnosti, že vozidlo je nuceno v místech s výrazným zakřivením zpomalit.

**Výstup č. 1**  
Určení rychlosti na základě  
třídy komunikace



**Výstup č. 2**  
Určení rychlosti na základě  
křivolakosti



**Výstup č. 3**  
Určení rychlosti na základě  
poloměru směrových oblouků



Obr. 29 Porovnání výstupů na úseku silnice

Tab. 11 Rozpětí intervalů pro jednotlivé výstupy

Časový interval	Výstup č. 1 třída komunikace (bez modelu)	Výstup č. 2 Křivolakost (model)	Výstup č. 3 poloměr oblouků (model)
8 min	2247	2058	1812
12 min	2127	1124	1713
16 min	2375	1136	847
20 min	2412	1459	1344

V Tab. 11 jsou uvedeny vzdálenosti hranic časových intervalů pro jednotlivé výstupy analýzy. Z výsledků je patrné, že v případě určení rychlosti podle třídy komunikace, která je pro celý zkoumaný úsek totožná, jsou vzdálenosti intervalů podobné. Stejně je tomu i v případě výstupu č. 2, kde je určena křivolakost pro celý úsek a tudíž se výsledné vzdálenosti příliš neliší. Při určení rychlosti pomocí poloměru oblouků je zaznamenáván velký výkyv hodnot ve třetím intervalu, který svoji polohou odpovídá soustavě velmi prudkých zatáček ve střední části úseku. To dokazuje fakt, že určení rychlosti na základě poloměru oblouků má potenciál větší přesnosti. Ovšem z důvodů zjištěných nedostatků v kapitole 3 a 6.2.1 je tento model stále nevyhovující.

## 7. ZÁVĚR

Pro implementaci atributu vyjadřujícího horizontální zakřivení komunikace do datového modelu silniční sítě existuje více možných postupů. V rámci bakalářské práce byl navržen postup, při kterém je využito geometrické složky vstupních dat pro výpočet úhlové změny pro jednotlivé segmenty linií tvořících úseky sítě. Hodnota úhlové změny je výchozí pro další analýzy průběhu komunikace, jako jsou výpočet křivolakosti a poloměru směrových oblouků. K jejímu získání byly použity výhradně metody dostupné v prostředí programu ESRI ArcGIS, což zajišťuje širokou použitelnost modelu. Neboť se jedná o běžně využívaný program pro správu geografických dat v soukromé sféře i státní správě v rámci České republiky.

Jedním z výsledků práce je návrh modelu pro výpočet křivolakosti komunikace. Aplikací na modelovém území bylo prokázáno, že křivolakost je vhodným ukazatelem vlastností komunikace a hodí se pro úpravu rychlosti průjezdu při analýzách silniční sítě. Díky tomu, že model nevykazuje žádné výrazné nepřesnosti, je vhodný pro přímou implementaci do praxe. Jedním z využití by mohlo být např. zpřesnění dojezdových časů pro jednotky IZS.

Dalším výsledkem je návrh postupu pro výpočet poloměru směrových oblouků v průběhu komunikace. Tento ukazatel je taktéž vhodný pro úpravu rychlosti průjezdu. Lze jej také použít při analýzách rizikových úseků. Ve stavu, ve kterém je model pro výpočet poloměru navržen, však vykazuje stále určitou nepřesnost. Je tedy na místě konstatovat, že model není vhodný pro přímou implementaci do praxe a poskytuje pouze návrh směru, jakým by se mohl vývoj takového modelu ubírat. Zdrojem nepřesností je způsob určování koncových bodů oblouků a eliminace zavádějících úhlových změn plynoucích z nedostatečné přesnosti vstupních dat. Zdokonalením metody by model mohl získat dostatečnou přesnost pro praktické využití.



## LITERATURA A POUŽITÉ ZDROJE

- [1] ARCTUR, D. - ZEILER, M. Designing Geodatabases: Case Studies in GIS Data Modeling. Redlands(California): ESRI, c1999-2004. ISBN 1-58948-021-X.
- [2] BULTER, J. Allison. Designing Geodatabases for Transportation. 1. California: Esri Press, 2008. 461 s. ISBN 978-1-58948-164-0.
- [3] ČSN 736101. Projektování silnic a dálnic. Praha: Český normalizační institut, 2004. Dostupné z WWW: <<http://www.fce.vutbr.cz/PKO/0M2/CSN736101-000-040.pdf>>
- [4] KŘIVDA, Vladislav, Michal RICHTÁŘ a Ivana OLIVKOVÁ. 2. *Silniční doprava: Učební text*. první. Ostrava: VŠB – TUO, 2007. ISBN 978-80-248-1521-3.
- [5] LONGLEY, P. A. - GOODCHILD, M. F. - MAGUIRE, D. J. - RHIND, D. W. Geographic Information Systems and Science. Chichester (England): Wiley and Sons, Ltd., c2001. ISBN 0-471-89275-0.
- [6] LOUTHAN, Michal. *Vztah digitálního modelu reliéfu a síťových analýz při řešení dopravních úloh*. Olomouc, 2010. 68 s. Diplomová práce. Univerzita Palackého v Olomouci.
- [7] LUTZ, Marc a David ASCHER. *Naučte se Python: Pohotová příručka*. Sebastopol, USA: O'reilly & Associates, Inc., 1999. ISBN 80-247-0367-X.
- [8] RASDORF, William, et al. Evaluation of GIS Applications for Horizontal Curve Data Collection. *Journal of Computing in Civil Engineering*. August 31, 2011, 8, s. 0-39. Dostupný také z WWW: <<http://online.sagepub.com/>>.
- [9] Segments of Circles. In: Ask Dr. Math: Math Forum [online]. Drexel University, 2012 [cit. 2012-04-07]. Dostupné z: <http://mathforum.org/dr.math/faq/faq.circle.segment.html#n1>
- [10] SLADKÝ, Jakub. *Síťové analýzy v GIS pro složky IZS*. Plzeň, 2009. 85 s. Diplomová práce. Západočeská univerzita v Plzni.
- [11] TUČEK, J. Geografické informační systémy: principy a praxe. 1. vydání. Praha: Computer Press, c1998. ISBN 80-7226-091-X.
- [12] VOŽENÍLEK, Vít. *Geografické informační systémy 1. : pojetí, historie, základní komponenty*. Olomouc: Vydavatelství Univerzity Palackého, 1998. 173 s. ISBN 80-7067-802-X.
- [13] Web-based Help [online]. 2008 [cit. 2011-04-19]. ArcGIS Resource Center. Dostupné z WWW: <<http://resources.arcgis.com/content/web-based-help>>.



## SEZNAM OBRÁZKŮ

- Obr. 1 Porovnání křivolakosti a indexu zakřivení
- Obr. 2 Dělení geografických dat (Voženílek 1998, upraveno)
- Obr. 3 Určení oblouků v průběhu linie
- Obr. 4 Kruhová úseč
- Obr. 5 Oblouk prostý kružnicový
- Obr. 6 Oblouk kružnicový s přechodnicemi (ČSN 736101 2004)
- Obr. 7 Převod oblouku kružnicového s přechodnicí na prostý kružnicový
- Obr. 8 Struktura víceúrovňového seznamu
- Obr. 9 Stanovení trendu
- Obr. 10 Výpočet úhlu k ose x
- Obr. 11 Úhlová změna s orientací
- Obr. 12 Postup výpočtu úhlu beta pro jednotlivé kombinace trendů
- Obr. 13 Struktura seznamu A
- Obr. 14 Field Calculator, výpočet poloměru oblouku
- Obr. 15 Rozdělení úseku na segmenty
- Obr. 16 Ukázka AT po dokončení druhé části modulu
- Obr. 17 Problematické úhly na koncích linií a při křížení
- Obr. 18 Struktura modelu
- Obr. 19 Vstupní část modelu
- Obr. 20 Struktura modelu, rozdělení segmentů
- Obr. 21 Struktura modelu, křížení a konce linií
- Obr. 22 Struktura modelu, úhlová změna
- Obr. 23 Ukázka AT po aplikaci Skriptu 3
- Obr. 24 Struktura modelu, výpočet poloměru oblouků

Obr. 25 Identifikované úhlové změny v průběhu linie

Obr. 26 Poloměry oblouků na vybraném úseku silnice

Obr. 27 Zastoupení poloměrů oblouků na území ORP Liberec pro jednotlivé třídy silnic

Obr. 28 Postup provedení síťové analýzy

Obr. 29 Porovnání výstupů na úseku silnice

## **SEZNAM TABULEK**

Tab. 1 Určení směrodatné rychlosti (Louthan 2009, s. 37)

Tab. 2 Určení směrodatné rychlosti (ČSN 736101 2004)

Tab. 3 Část výpočtu Newtonovou metodou (Segments of Circles 2012)

Tab. 4 Doporučené délky přechodnic (ČSN 736101 2004)

Tab. 5 Vlastnosti nových polí AT

Tab. 6 Křivolakost jednotlivých tříd silnic na území ORP Liberec

Tab. 7 Úprava rychlosti podle křivolakosti

Tab. 8 Určení průměrné rychlosti

Tab. 9 Doporučené poloměry směrových oblouků v závislosti na rychlosti a dostředném sklonu (ČSN 2004)

Tab. 10 Výsledky analýzy pro určené body (hodnoty jsou uvedeny v minutách)

Tab. 11 Rozpětí intervalů pro jednotlivé výstupy

## **SEZNAM PŘÍLOH**

- Příloha 1. Skript 1 – Určení úhlu a orientace
- Příloha 2. Skript 2 – Úprava koncových úhlů
- Příloha 3. Skript 3 – Řazení segmentů k obloukům
- Příloha 4. Analýza časové dostupnosti v rámci území ORP Liberec na základě atributu třídy komunikace
- Příloha 5. Analýza časové dostupnosti v rámci území ORP Liberec na základě atributu křivolakosti
- Příloha 6. Analýza časové dostupnosti v rámci území ORP Liberec na základě atributu poloměru směrového oblouku

## Příloha 1. Skript 1 – Určení úhlu a orientace

```
import arcpy, locale, math
locale.setlocale(locale.LC_ALL, '')
originalinputfeature = arcpy.GetParameterAsText(0)
outputfeature = arcpy.GetParameterAsText(1)
arcpy.CopyFeatures_management(originalinputfeature, outputfeature)
arcpy.AddField_management(outputfeature,"UhelMinus","DOUBLE","","","","UhelSegment-1")
arcpy.AddField_management(outputfeature,"UhelPlus","DOUBLE","","","","UhelSegment+1")
arcpy.AddField_management(outputfeature,"DirectionMinus","TEXT","","","","")
arcpy.AddField_management(outputfeature,"DirectionPlus","TEXT","","","","")
desc = arcpy.Describe(outputfeature)
cur = arcpy.SearchCursor(outputfeature)
s=[0.0,0.0]
e=[0.0,0.0]
l=[s,e]
D = [l]
for row in cur:
    feat=row.shape
    firstpoint = feat.firstPoint
    lastpoint = feat.lastPoint
    startx = firstpoint.X
    starty = firstpoint.Y
    endx = lastpoint.X
    endy = lastpoint.Y
    s=[startx,starty]
    e=[endx,endy]
    l=[s,e]
    D.append (l)
del row
del cur
beta = 0
direction = "0"
startx1 = D[1][0][0]
starty1 = D[1][0][1]
endx1 = D[1][1][0]
endy1 = D[1][1][1]
cur = arcpy.UpdateCursor (outputfeature)
for row in cur:
    IDS = row.getValue(desc.OIDFieldName)
    row.UhelMinus = beta
    row.DirectionMinus = direction
    try:
        del beta, direction, trend1, trend2, alfa1, alfa2
```

```

except:
beta = []
try:
    startx2=D[IDS+1][0][0]
    starty2=D[IDS+1][0][1]
    endx2=D[IDS+1][1][0]
    endy2=D[IDS+1][1][1]
except:
    startx2=D[IDS][0][0]
    starty2=D[IDS][0][1]
    endx2=D[IDS][1][0]
    endy2=D[IDS][1][1]
if startx1==endx1 and starty1<endy1:
    trend1=1
elif startx1!=endx1 and starty1!=endy1 and startx1<endx1 and starty1<endy1:
    trend1=2
elif starty1==endy1 and startx1<endx1:
    trend1=3
elif startx1!=endx1 and starty1!=endy1 and startx1<endx1 and starty1>endy1:
    trend1=4
elif startx1==endx1 and starty1>endy1:
    trend1=5
elif startx1!=endx1 and starty1!=endy1 and startx1>endx1 and starty1>endy1:
    trend1=6
elif starty1==endy1 and startx1>endx1:
    trend1=7
else:
    trend1=8
if trend1 in [1,5]:
    alfa1=90.0
elif trend1 in [3,7]:
    alfa1=0.0
elif trend1 in [2,4,6,8]:
    a1=abs(startx1-endx1)
    b1=abs(starty1-endy1)
    try:
        alfa1=math.degrees(math.atan(b1/a1))
    except:
        alfa1=0
if startx2==endx2 and starty2<endy2:
    trend2=1
elif startx2!=endx2 and starty2!=endy2 and startx2<endx2 and starty2<endy2:
    trend2=2
elif starty2==endy2 and startx2<endx2:

```

```

trend2=3
elif startx2!=endx2 and starty2!=endy2 and startx2<endx2 and starty2>endy2:
    trend2=4
elif startx2==endx2 and starty2>endy2:
    trend2=5
elif startx2!=endx2 and starty2!=endy2 and startx2>endx2 and starty2>endy2:
    trend2=6
elif starty2==endy2 and startx2>endx2:
    trend2=7
else:
    trend2=8
if trend2 in [1,5]:
    alfa2=90.0
elif trend2 in [3,7]:
    alfa2=0.0
elif trend2 in [2,4,6,8]:
    #delka na ose x
    a2=abs(startx2-endx2)
    b2=abs(starty2-endy2)
    try:
        alfa2=math.degrees(math.atan(b2/a2))
    except:
        alfa2=0
if trend1==trend2:
    beta=abs(alfa1-alfa2)
elif trend1 in [1,3,5,7] and trend2 in [1,3,5,7]:
    beta=abs(alfa1-alfa2)
elif abs(trend1-trend2) in [0,1,2,6,7]:
    if (trend1 in [2,4] and trend2 in [6,8])or(trend1 in [6,8] and trend2 in [2,4]):
        beta=abs(180.0-(alfa1+alfa2))
    elif (trend1 in [8,2] and trend2 in [4,6])or(trend1 in [4,6] and trend2 in [8,2]):
        beta=abs(alfa1+alfa2)
    elif (trend1 in [2,4] and trend2 in [2,4])or(trend1 in [6,8] and trend2 in [6,8]):
        beta=abs(alfa1+alfa2)
    #kombinace uhlu v 1,3,5,7 s uhly 2,4,6,8 a naopak
    elif (trend1 in [1,5] and trend2 in [2,4,6,8])or(trend1 in [2,4,6,8] and trend2 in
[1,5]):
        beta=abs(alfa1-alfa2)
    elif (trend1 in [3,7] and trend2 in [2,4,6,8])or(trend1 in [2,4,6,8] and trend2 in
[3,7]):
        beta=abs(alfa1+alfa2)
    else:
        beta = 90.0
else:
    beta = 90.0

```

```

print "vysledny uhel pro segmenty je:",beta
#zmena smeru uhlu (Ssegmenty)
if trend1==trend2 and alfa1==alfa2:
    direction="0"
elif trend1==trend2 and trend1 in [2,6]:
    if alfa1>alfa2:
        direction="right"
    if alfa1<alfa2:
        direction="left"
elif trend1==trend2 and trend1 in [4,8]:
    if alfa1>alfa2:
        direction="left"
    if alfa1<alfa2:
        direction="right"
elif (trend1<trend2) or (trend1==8 and trend2==1) or (trend1==7 and trend2==1) or
(trend1==8 and trend2==2):
    if (trend1==1 and trend2==8) or (trend1==1 and trend2==7) or (trend1==2 and trend2==8):
        direction="left"
    else:
        direction="right"
elif trend1>trend2 or (trend1==1 and trend2==8) or (trend1==1 and trend2==7) or (trend1==2
and trend2==8):
    direction="left"
else:
    direction="right"
if beta < 0.1:
    beta=0
    direction="0"
startx1 = startx2
starty1 = starty2
endx1 = endx2
endy1 = endy2
row.UhelPlus = beta
row.DirectionPlus = direction
cur.updateRow(row)
del cur, row
print "FINISH"

```

## Příloha 2. Skript 2 – Úprava koncových úhlů

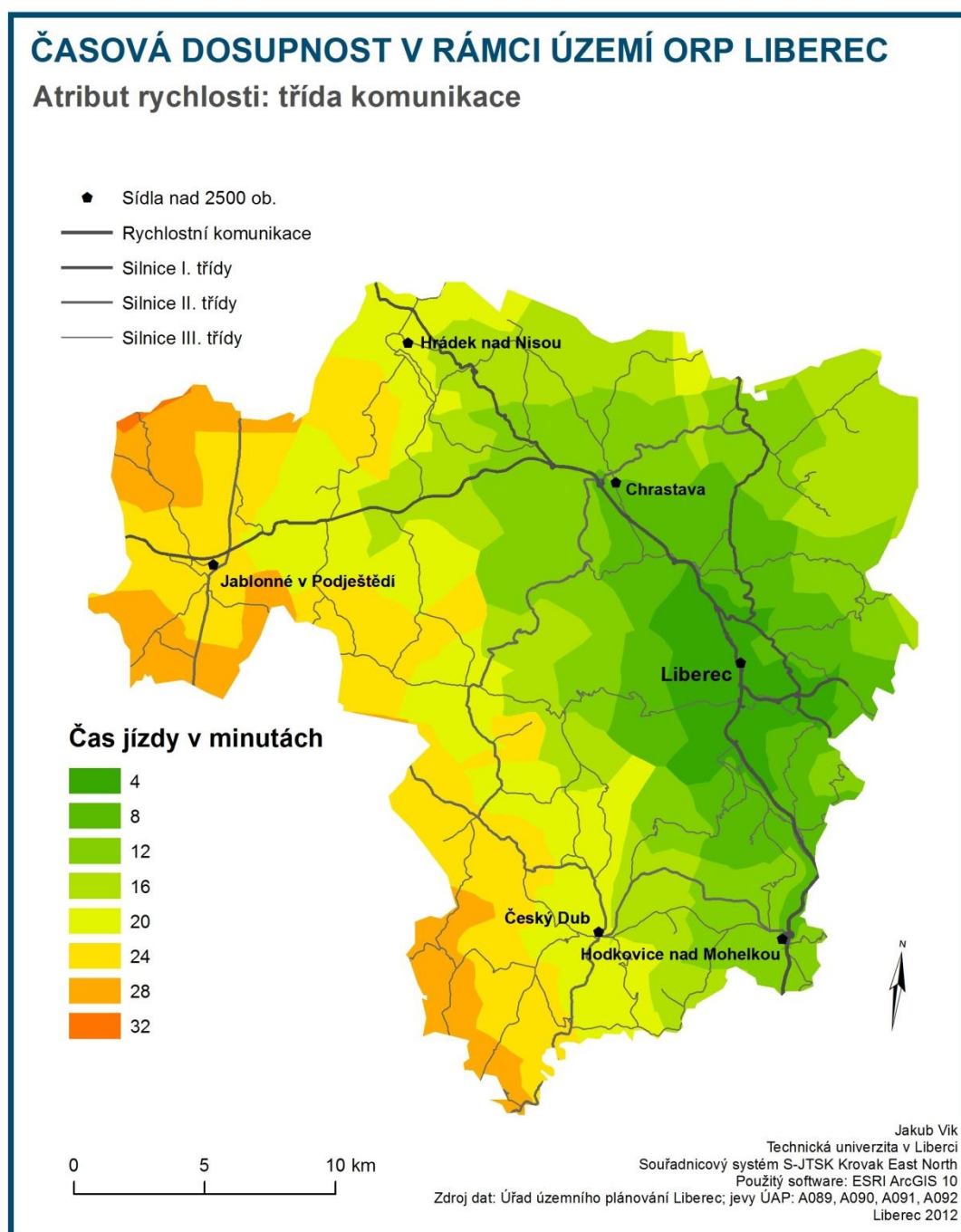
```
import arcpy, math
originalinputfeature = arcpy.GetParameterAsText(0)
outputfeature = arcpy.GetParameterAsText(1)
arcpy.CopyFeatures_management(originalinputfeature, outputfeature)
cur = arcpy.SearchCursor (outputfeature)
A = [1]
for row in cur:
    if row.END == "end" or row.END == "crossroad":
        A.append (1)
    else:
        A.append (0)
del cur, row
A.append(1)
cur = arcpy.UpdateCursor (outputfeature)
FID = 1
for row in cur:
    if A[FID]==1:
        if A[FID-1]==1:
            row.UhelMinus = 0
            row.DirectionMinus = "0"
            cur.updateRow(row)
        if A[FID+1]==1:
            row.UhelPlus = 0
            row.DirectionPlus = "0"
            cur.updateRow(row)
    FID = FID + 1
del cur, row
print "FINISH"
```



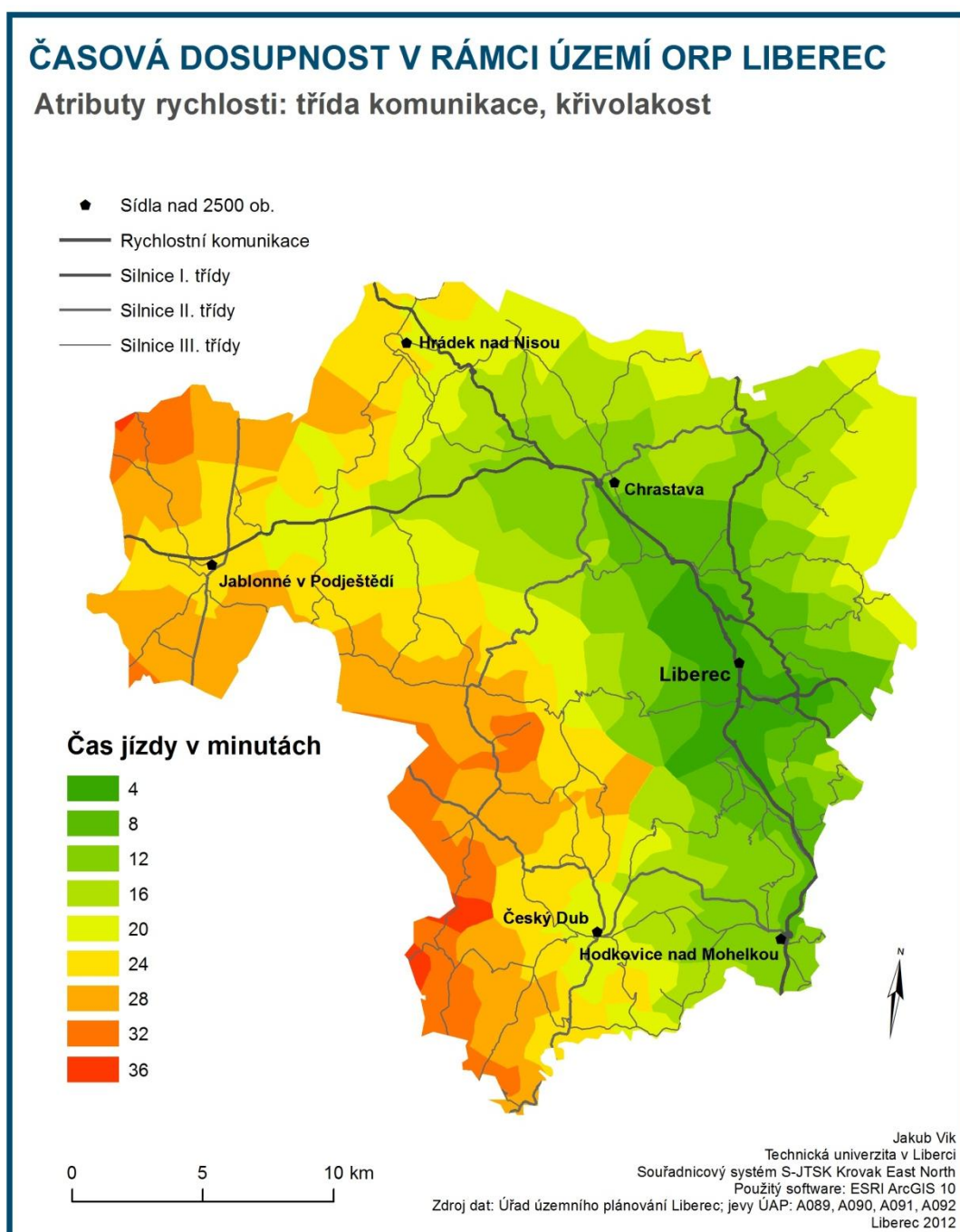
### Příloha 3. Skript 3 – Řazení segmentů k obloukům

```
import arcpy, locale, math
locale.setlocale(locale.LC_ALL, '')
originalinputfeature = arcpy.GetParameterAsText(0)
outputfeature = arcpy.GetParameterAsText(1)
arcpy.CopyFeatures_management(originalinputfeature, outputfeature)
arcpy.AddField_management(outputfeature, "OBLOUK", "DOUBLE", "", "", "", "")
cur = arcpy.UpdateCursor (outputfeature)
Curve=0
CurveNumber=1
for row in cur:
    if Curve==0:
        if row.UhelPlus != 0:
            Curve=1
            row.OBLOUK = CurveNumber
            cur.updateRow(row)
        else:
            Curve=Curve
    else:
        if row.UhelPlus == 0:
            row.OBLOUK = CurveNumber
            cur.updateRow(row)
            Curve=0
            CurveNumber=CurveNumber+1
        else:
            row.OBLOUK = CurveNumber
            cur.updateRow(row)
del cur, row
```

Příloha 4. Analýza časové dostupnosti v rámci území ORP Liberec na základě atributu třídy komunikace



Příloha 5. Analýza časové dostupnosti v rámci území ORP Liberec na základě atributu křivolakosti



Příloha 6. Analýza časové dostupnosti v rámci území ORP Liberec na základě atributu poloměru směrového oblouku

